AFIT/DS/ENY/96-6

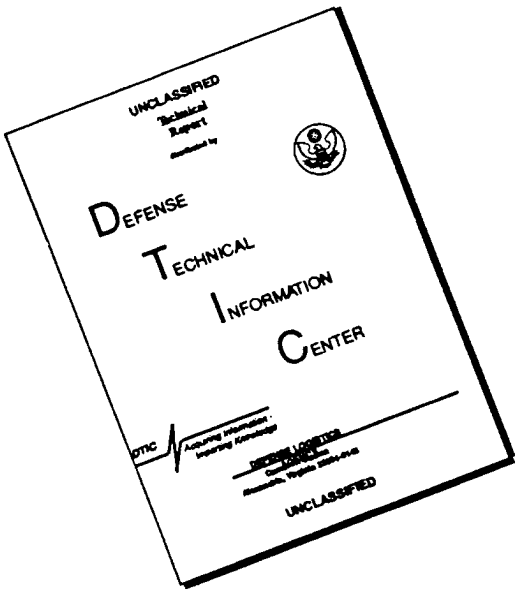
NONLINEAR ANALYSIS OF AIRFOIL FLUTTER AT TRANSONIC SPEEDS

DISSERTATION

Scott A. Morton
Captain, USAF

AFIT/DS/ENY/96-6

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

May 1996 Dissertation

NONLINEAR ANALYSIS OF AIRFOIL FLUTTER AT TRANSONIC SPEEDS

Scott A. Morton

Air Force Institute of Technology, WPAFB OH 45433-6583

AFIT/DS/ENY/96-6

Dr Joe Shang
WL/FIM Wright-Patterson AFB, Oh 45433

Hopf-bifurcation analysis is used to determine flutter boundaries, for variations of Mach number and selected structural parameters, of a pitch and plunge airfoil (PAPA) at transonic Mach number conditions. The PAPA model is a coupling of the Euler equations and a two-degree-of-freedom structural model with either linear or nonlinear elements. The Euler equations are discretized using an upwind total variation diminishing scheme (TVD) of Harten and Yee. Equilibrium solutions of the PAPA model are computed using Newton's method and dynamic solutions are explicitly integrated in time with first-order accuracy. The Hopf-bifurcation point, which models the flutter condition, is computed using a modified form of the Griewank and Reddien algorithm. A path of Hopf-points is computed as a function of Mach number to produce a Mach flutter boundary. The flutter boundary is validated by time integration of the PAPA model. Also, flutter boundaries are obtained through variation of additional aerodynamic and structural parameters.

Hopf Bifurcation, Transonic, Aeroelasticity, Newton's Method, TVD 210

UNCLASSIFIED UNCLASSIFIED UNCLASSIFIED UL

AFIT/DS/ENY/96-6

NONLINEAR ANALYSIS OF AIRFOIL FLUTTER AT TRANSONIC SPEEDS

DISSERTATION

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

Scott A. Morton

Captain, USAF

Sponsored by the Flight Dynamics Directorate, Wright Laboratory, WPAFB, OH.

May, 1996

AFIT/DS/ENY/96-6

NONLINEAR ANALYSIS OF AIRFOIL FLUTTER AT TRANSONIC SPEEDS

Scott A. Morton

Captain, USAF

Approved:

PHILIP S. BERAN, Research Advisor
Professor
Department of Aeronautics and Astronautics

THOMAS A. BUTER, Committee Member
Major, USAF
Department of Aeronautics and Astronautics

DAVID L. COULLIETTE, Committee Member
Lt Col, USAF
Department of Mathematics and Statistics

JOSEPH J.S. SHANG, Committee Member
Adjunct Professor
Department of Aeronautics and Astronautics

EDWARD F. MYKYTKA, Dean's Representative
Deputy Department Head
Department of Operational Sciences

Robert A. Calico, Jr.
Dean, Graduate School of Engineering

Table of Contents

	Page
List of Figures	iv
List of Tables	v
List of Symbols	vi
Abstract	xii
I. Introduction	1-1
1.1 Historical Perspective on Flutter	1-2
1.1.1 Summary of Experimental Data	1-2
1.1.2 Summary of Numerical Methods	1-5
1.1.3 Nonlinear Structural Models	1-14
1.2 Research Scope and Objectives	1-15
1.2.1 Research Objectives	1-16
1.2.2 Research Solution Method	1-17
1.2.3 Research Overview	1-20
II. Functional Analysis of Flows with Shocks	2-1
2.1 Elements of Functional Analysis	2-1
2.2 Hopf-Bifurcation System	2-5
2.3 Newton's Method	2-7
III. Transonic Flows About Fixed Airfoils	3-1
3.1 Governing Equations	3-1
3.2 General Coordinate Transformation	3-3
3.3 Time-Integration Scheme	3-3
3.4 Computational Grid	3-5

Page

3.5 Boundary Conditions 3-8

3.5.1 Farfield Inflow Boundary Conditions 3-9

3.5.2 Farfield Outflow Boundary Conditions 3-10

3.5.3 Airfoil Surface Boundary Conditions 3-10

3.5.4 Cut Boundary Conditions 3-13

3.6 Calculation of Force and Moment Coefficients 3-14

3.7 Equilibrium System 3-15

IV. Static Airfoil Results 4-1

4.1 Algorithm Validation (Equilibrium Solutions) 4-1

4.1.1 Grid-Sensitivity Study 4-1

4.1.2 Comparison with AGARD Solution 4-6

4.2 Convergence Properties of Newton-TVD and TVD-Time-Integration Algorithms 4-7

4.2.1 Newton-TVD and TVD-Time-Integration Comparison 4-7

4.2.2 Variation of Numerical Jacobian Parameter 4-9

4.3 Variation of Mach Number and Angle-of-Attack 4-9

4.3.1 NACA 0012 4-10

4.3.2 NACA 64A006 4-11

V. Airfoils With Structural Coupling 5-1

5.1 Fluid-Structure Interaction System 5-1

5.1.1 Pitch and Plunge Structural Model 5-2

5.1.2 Moving Airfoil Aerodynamics Model 5-6

5.1.3 PAPA Time-Integration Algorithm 5-11

5.1.4 PAPA Equilibrium Algorithm 5-11

5.2 PAPA Validation 5-13

5.2.1 Structural Model Validation 5-14

5.2.2 Forced Oscillation Validation 5-16

Page

5.2.3 PAPA Time-Integration Algorithm Validation 5-18

5.2.4 PAPA Equilibrium and Time-Integration Algorithm Consistency 5-19

5.3 NACA 64A006 PAPA Equilibrium Solution Space 5-22

5.3.1 Static Pretwist Effects 5-22

5.3.2 Mach Number Effects 5-22

VI. Hopf-Bifurcation Analysis 6-1

6.1 General Framework for Hopf-Bifurcation Analysis 6-1

6.2 Griewank and Reddien Algorithm 6-3

6.2.1 Development of GR Algorithm 6-3

6.2.2 Summary of GR Algorithm 6-5

6.2.3 Limitations of GR Algorithm 6-6

6.3 Blocked Gauss-Seidel Newton Algorithm 6-6

6.3.1 BGSN5 Algorithm Implementation 6-8

6.3.2 BGSN5 Algorithm Summary 6-13

6.3.3 BGSN4 Algorithm Summary 6-14

VII. PAPA Results 7-1

7.1 Hopf-Bifurcation Validation 7-1

7.1.1 PAPA Eigenvalue Analysis 7-3

7.1.2 PAPA Time-Integration Analysis 7-4

7.1.3 Hopf-Bifurcation Point Sensitivity to Grid Resolution 7-5

7.2 BGSN4/BGSN5 Convergence Properties 7-7

7.3 Hopf-Point Results 7-10

7.3.1 Mach Flutter Boundary 7-14

7.3.2 Damping Flutter Boundary 7-19

7.3.3 Static Pretwist Flutter Boundary 7-20

Page

VIII. Nonlinear Structural Models 8-1

8.1 Equations of Motion 8-1

8.2 Nonlinear Structural Model Validation 8-4

8.3 Nonlinear Structural Model Hopf-Point Results 8-6

IX. Conclusions and Recommendations 9-1

9.1 Summary and Conclusions 9-1

9.1.1 Objective 1 9-1

9.1.2 Objective 2 9-3

9.1.3 Objective 3 and Objective 4 9-3

9.1.4 Objective 5 9-5

9.1.5 Objective 6 9-5

9.1.6 Objective 7 9-7

9.2 Recommendations for Further Research 9-9

Appendix A. Harten-Yee Total Variation Diminishing Scheme A-1

A.1 One-Dimensional TVD Scheme A-1

A.2 One-dimensional TVD Model Problem A-3

A.2.1 Governing Equations A-4

A.2.2 Application of Numerical Scheme A-5

A.2.3 Results and Conclusions A-7

A.3 Two-Dimensional TVD Scheme A-10

Appendix B. Numerical Jacobian B-1

B.1 Regular Point Jacobian Matrices B-1

B.2 Hopf-Point Jacobian Matrices B-3

Appendix C. Bordering Algorithm C-1

Page

Appendix D. TVDntiAE Run Matrix and Grid Specification D-1

D.1 Run Matrix and Grid Specification Tables D-1

D.2 Computer Usage Estimate D-1

D.3 BGSN5 and BGSN4 Profile Listing D-5

Appendix E. TVDntiAE Code Summary E-1

E.1 Code Overview E-1

E.2 List of Subroutines E-2

E.3 PACLIB Overview E-4

E.4 Data Archival Structure E-5

Bibliography BIB-1

List of Figures

Figure		Page
1.1.	The Effect of Violent Flutter, Theodorsen and Garrick [130]	1-3
2.1.	Examples of a Limit Point, and Sub- and Supercritical Pitchfork Bifurcation Points	2-2
2.2.	Solution Path With A Hopf-Point	2-4
2.3.	Solution Path With A Bistable Hopf-Point	2-4
3.1.	Computational Domain and Index Definition	3-4
3.2.	Trailing-Edge Schematic	3-7
3.3.	Representative Grid (NACA 64A006 Airfoil)	3-7
3.4.	Node Distribution Near NACA 64A006 Airfoil (Representative Grid)	3-8
3.5.	Relationship Between the Physical and Computational Domains	3-9
3.6.	Grid Point-Surface Geometry	3-11
4.1.	Sensitivity to Outer Domain Size	4-3
4.2.	Sensitivity to Wall Spacing	4-4
4.3.	Sensitivity to Node Quantity in Normal Direction	4-4
4.4.	Sensitivity to Node Quantity Around the Airfoil	4-5
4.5.	Surface Cp: $M_\infty = 0.8$ and $\alpha = 1.25^o$	4-7
4.6.	Convergence Histories of Newton-TVD and TVD-Time-Integration Algorithms	4-8
4.7.	Convergence Histories for Various Numerical Jacobian Accuracies (Newton-TVD)	4-10
4.8.	Effects of Freestream Mach Number on Mach Contours: NACA 0012, $\alpha = 1.25^o$	4-12
4.9.	Effects of Angle-of-Attack on Mach Contours: NACA 0012, $M_\infty = 0.80$	4-13
4.10.	Effects of Angle-of-Attack on Mach Contours: NACA 64A006, $M_\infty = 0.85$	4-14
4.11.	Variation of Angle-of-Attack: NACA 64A006, $M_\infty = 0.85$	4-15
5.1.	Airfoil Structural Coupling Model	5-2
5.2.	Inertial Reference Frame	5-4

Figure		Page
5.3.	Dynamic Airfoil Algorithm Flow Chart	5-11
5.4.	Structural Model Validation for Harmonic Motion in α and h	5-15
5.5.	NACA 64A006 Forced Pitch Comparison: $M_\infty = 0.87$	5-17
5.6.	NACA 64A006 Forced Plunge Comparison: $M_\infty = 0.87$	5-18
5.7.	Comparison of Time-Integration Methods: $M_\infty = 0.87$, $\alpha_0 = 0^o$, and $\zeta_\alpha = \zeta_h = 0$	5-20
5.8.	Overdamped Time Integration to Steady-State	5-21
5.9.	Equilibrium Angle-of-Attack: $M_\infty = 0.80$	5-23
5.10.	Equilibrium Angle-of-Attack: $\alpha_0 = 1.25^o$	5-24
7.1.	Eigenvalue Migration and Associated Hopf Curve: $\bar{u} = 6.5 - 8.5$	7-5
7.2.	Hopf-Points for Various Grids	7-7
7.3.	Convergence Histories for Newton's Method and Various Approximations to Newton's Method	7-8
7.4.	Convergence Histories for BGSN5 and BGSN4	7-10
7.5.	Time-Integration of PAPA Model: $\bar{u} = 9.0$ and 10.29 and $M_\infty = 0.85$	7-14
7.6.	Time-Integration of PAPA Model: $\bar{u} = 11.0$ and 14 and $M_\infty = 0.85$	7-14
7.7.	Hopf-Bifurcation Curve: $M_\infty = 0.85$, $\alpha_0 = 0^o$, and $\zeta_\alpha = \zeta_h = 0.5$	7-15
7.8.	Mach Flutter Boundary: $\alpha_0 = 0^o$	7-16
7.9.	Time-Integration: $\alpha_0 = 0^o$ and $\zeta_\alpha = \zeta_h = 0.5$	7-17
7.10.	Mach Contours for a Limit-Cycle Oscillation: $M_\infty = 0.85$, $\bar{u} = 14$	7-18
7.11.	Pitch Time-History: $M_\infty = 0.85$, $\bar{u} = 14$	7-18
7.12.	Pitch and Plunge Damping Flutter Boundary: $M_\infty = 0.85$ and $\alpha_0 = 0^o$	7-19
7.13.	Static Pretwist Flutter Boundary: $M_\infty = 0.85$ and $\zeta_\alpha = \zeta_h = 0.5$	7-20
7.14.	Time-Integration: $\alpha_0 = 0.5^o$, $\bar{u} = 11.25$ and $\bar{u} = 11.75$	7-21
8.1.	Bilinear Torsion Model	8-3
8.2.	Nonlinear Structural Model Validation for Harmonic Motion in α and h	8-5
8.3.	Hopf Curve: $\alpha_s = 0.5^o$, $M_\infty = 0.85$	8-8
8.4.	Flutter Boundaries: $M_\infty = 0.85$	8-8

Figure Page

8.5. Flutter Boundary: $f_s/\alpha_s = 0.5 - 2.0$, $M_\infty = 0.85$ 8-9

A.1. Quasi-One Dimensional Nozzle Geometry A-5

A.2. Convergence History of QSHKNEW (Case 2) A-10

A.3. Convergence History of QSHKTIM (Case 2) A-11

A.4. Grid Sensitivity Study: 20 Nodes (Case 1) A-11

A.5. Grid Sensitivity Study: 50 Nodes (Case 2) A-12

C.1. Matrix Structure C-2

List of Tables

Table — Page

1.1. Index of References for Experimental and Numerical Data extracted from [40] (*-interacted boundary layer model, +-nonisentropic corrections, TSD-transonic small disturbance equation, FP-full potential equation, EE-Euler equations, NS-Navier-Stokes equations) 1-5

3.1. Grid Definitions 3-6

4.1. Case Definitions for Sensitivity Analysis 4-2

4.2. Summary of Sensitivity Evaluations 4-6

4.3. Convergence Properties Run Summary 4-8

4.4. Run Summary: Variation of M_∞ and α 4-11

5.1. Run Summary: PAPA Validation 5-14

5.2. Structural Model Integration Error Norms 5-15

7.1. Run Summary: Hopf-Point Validation (Eig-eigenvalue calculation, TI-time-integration, BGSN4-Hopf-point with BGSN4 algorithm, BGSN5-Hopf-point with BGSN5 algorithm) 7-2

7.2. Eigenvalue with Maximum Real Part for Various Reduced Velocities 7-4

7.3. Period of Oscillation Comparison for Various Reduced Velocities ($T_H = 58.23$ at $\bar{u}^* = 6.937$ 7-5

7.4. Run Summary: Grid Sensitivity 7-6

7.5. Run Summary: Hopf-Point Results 7-11

7.6. Computational Resource Requirements (DEC 4620/Alpha 150 Mhz Workstation). Note: $*$ denotes estimated. 7-13

8.1. Structural Model Validation Error Norms 8-5

8.2. Error Norms for a Variation in Bilinear Ratio: $\Delta t = 0.1$ 8-6

8.3. Run Summary: Nonlinear Structural Hopf-Point Results 8-6

Table Page

A.1. Summary of Computer Runs for QSHKTIM and QSHKNEW A-9

A.2. Convergence and Accuracy Summary of Cases Run A-10

B.1. Comparison of Numerical versus Analytic Jacobian Matrices B-2

D.1. Grid Definitions (Table 3.1) D-2

D.2. Case Definitions for Sensitivity Analysis (Table 4.1) D-3

D.3. Convergence Properties Run Summary (Table 4.3) D-3

D.4. Run Summary: PAPA Validation (Table 5.1) D-3

D.5. Run Summary: Hopf-Point Validation (Table 7.1) D-4

D.6. Run Summary: Grid Refinement (Table 7.4) D-4

D.7. Run Summary: Hopf-Point Results (Table 7.5) D-4

D.8. Run Summary: Nonlinear Structural Hopf-Point Results (Table 8.3) D-4

D.9. Profile listing for BGSN4: generated Thu Apr 25 05:06:26 1996, each sample covers 4.00 byte(s) for 0.0003% of 320.4619 seconds D-6

D.10. Profile listing for BGSN5: generated Thu Apr 25 06:33:38 1996, each sample covers 4.00 byte(s) for 0.0003% of 320.4619 seconds D-7

List of Symbols

The following list of terms defines the variables used in this document.

A ..... Jacobian matrix of the aerodynamic equations away from the cut (banded)

A_{bw} ..... bandwidth of a matrix A

α ..... angle-of-attack

α_0 ..... static pretwist (or wing root angle-of-attack)

α_{eq} ..... equilibrium angle-of-attack

α_{LC} ..... peak angle-of-attack in an LCO

$\hat{\alpha}$ ..... pitch forcing function amplitude

B ..... columns of F_U related to cut variables

b ..... half chord

β ..... eigenvalue, also known as complex frequency

β_{max} ..... eigenvalue with the largest real part

C ..... rows of F_U related to cut equations

C_d ..... coefficient of drag

C_l ..... coefficient of lift

C_m ..... coefficient of moment, positive nose up

C_p ..... coefficient of pressure

c_p ..... specific heat for constant pressure

c_v ..... specific heat for constant volume

c ..... chord length

D ..... matrix of F_U elements related to cut equations and cut variables

D_h ..... structural damping coefficient in plunge

D_α ..... structural damping coefficient in pitch

$\Delta wall$ ..... spacing between $j = 1$ and $j = 2$ nodes

Δte ..... trailing edge truncated portion

δ_1 ..... TVD entropy correction parameter

ϵ ..... perturbation parameter

ϵ_{jac} ..... numerical Jacobian parameter

E_t ..... total energy

e ..... internal energy

$\hat{\mathbf{e}}_1$ ..... airfoil fixed unit basis vector aligned with the chord

$\hat{\mathbf{e}}_2$ ..... airfoil fixed unit basis vector normal to the chord

F ..... nonlinear system of aerodynamic equations

$\mathcal{F}$ ..... nonlinear Hopf-bifurcation system

$\bar{F}$ ..... Euler equation fluxes in the x direction

$\hat{F}$ ..... Euler equation fluxes in the ξ direction

$\tilde{\hat{F}}$ ..... modified TVD flux in the ξ direction

F_U ..... Jacobian matrix of aerodynamic equations

G ..... system of aerodynamic and structural equations

G_Y ..... Jacobian matrix of G

$\bar{G}$ ..... Euler equation fluxes in the y direction

$\hat{G}$ ..... Euler equation fluxes in the η direction

$\tilde{\hat{G}}$ ..... modified TVD flux in the η direction

γ ..... ratio of specific heats

η ..... transformed coordinate in the vertical direction

h ..... plunge magnitude

$\hat{h}$ ..... plunge forcing function amplitude

$\mathbf{i}$ ..... $\sqrt{-1}$

i ..... ξ index in the computational space

$\hat{\mathbf{i}}$ ..... inertial unit basis vector in the x direction

I ..... maximum i index

I_α ..... mass moment of inertia about the elastic axis

j ..... η index in the computational space

$\mathcal{J}$ ..... Jacobian of the transformation metrics

$\hat{\mathbf{j}}$ ..... inertial unit basis vector in the y direction

J ..... maximum j index

K_h ..... plunge spring stiffness coefficient

K_α ..... pitch spring stiffness coefficient

$\mathcal{K}$ ..... structural model equations

k ..... collocation index for nodes

$\tilde{\lambda}$ ..... free parameter of a nonlinear system, $G = 0$

L_η ..... TVD operator in the η direction

L_ξ ..... TVD operator in the ξ direction

LCO ..... limit cycle oscillation

M ..... Mach number

m ..... airfoil mass per unit span

μ_s ..... airfoil mass ratio

N ..... number of equations in the nonlinear system

N_{cut} ..... number of equations in the nonlinear system related to the cut

ν ..... iterate number

ν_{cfl} ..... Courant number

$\mathbf{\Omega}$ ..... pitch rate vector

Ω ..... pitch rate magnitude

ω_α ..... pitch natural frequency

ω_h ..... plunge natural frequency

P ..... complex eigenvector of G_Y

P_1 ..... real component of P

P_2 ..... imaginary component of P

p ..... pressure

Q ..... structural equation source vector

q ..... normalization vector

R_{max} ..... radius of outer domain

$\mathbf{r_b}$ ..... inertial position vector of a point on the airfoil

$\mathbf{r_0}$ ..... inertial position vector of the elastic axis

r ..... mass static offset

r_α ..... radius of gyration

ρ ..... density

ρu ..... u momentum

ρv ..... v momentum

S ..... structural dependent variables of $\mathcal{K} = 0$

t ..... time

Δt ..... time step

T_p ..... period of a time-dependent system

T ..... temperature

Θ ..... imaginary component of the eigenvalue, β

$\bar{\Theta}$ ..... local slope of the airfoil

$\bar{U}$ ..... conserved dependent variables at a node

$\hat{U}$ ..... transformed conserved dependent variables at a node

U ..... concatenation of all conserved dependent variables

U_{bn} ..... velocity of a point on the surface in the normal direction

u ..... inertial velocity in the x direction

u_g ..... inertial grid velocity in the x direction

u_e ..... velocity component in the $\hat{\mathbf{e}}_1$ direction

u' ..... velocity component tangent to the surface

$\bar{u}$ ..... reduced velocity

$\bar{u}^*$ ..... critical reduced velocity

v ..... inertial velocity in the y direction

v_g ..... inertial grid velocity in the y direction

v_e ..... velocity component in the $\hat{\mathbf{e}}_2$ direction

v' ..... velocity component normal to the surface

V_∞ ..... freestream velocity

$\mathcal{V}$ ..... contravariant v velocity

x ..... coordinate in the $\hat{\mathbf{i}}$ direction

x_{cg} ..... center of gravity position along the chord line

x_α ..... nondimensional mass static offset, positive cg forward of ea

$\bar{x}$ ..... horizontal distance from the elastic axis to a point on the airfoil

ξ ..... transformed coordinate in the horizontal direction of the computational domain

$\mathcal{X}$ ..... dependent variables of the Hopf-bifurcation system

Y ..... dependent variables of a nonlinear system, $G = 0$

y_{cg} ..... center of gravity position above the chord line

$\bar{y}$ ..... vertical distance from the elastic axis to a point on the airfoil

ζ_h ..... plunge damping ratio

ζ_α ..... pitch damping ratio

Superscripts

^{n} ..... time step index

Subscripts

$_b$ ..... body

eq ..... equilibrium

in ..... inertial

∞ ..... freestream

Abstract

Hopf-bifurcation analysis is used to determine flutter boundaries of a pitch and plunge airfoil (PAPA) at transonic Mach number conditions. The PAPA model is a coupling of the Euler equations and a two-degree-of-freedom structural model composed of linear and torsional springs. The Euler equations are discretized using an upwind total variation diminishing scheme (TVD) of Harten and Yee. Equilibrium solutions of the PAPA model are computed using Newton's method and dynamic solutions are explicitly integrated in time with first-order accuracy. The Hopf-bifurcation point, which models the flutter condition, is computed directly by solving an extended system of equilibrium equations following the approach of Griewank and Reddien. The extended system is solved using a blocked Gauss-Seidel Newton relaxation scheme to improve computational resource requirements.

Hopf-points are computed and validated through consistency checks with time-integration solutions, eigenvalue analysis of the equilibrium system Jacobian matrix, and solutions from the open literature. Time-integration allows time-dependent behavior of the PAPA system to be analyzed near the stability boundary and allows for comparison of limit-cycle oscillations. Paths of Hopf-points are computed for variations in Mach number, structural damping, and static pretwist. The Hopf-point computation is shown to have a factor of ten performance advantage over the time-integration method for a single Hopf-point and even greater performance advantage when computing flutter boundaries with continuation.

The PAPA model is generalized to include a bilinear torsional spring. The bilinear torsional spring is in a two-parameter family of nonlinear structural models which includes freeplay. Flutter boundaries are computed for a wide range of the two parameters. A combination of the two parameters yields a single flutter boundary in terms of a single normalization variable.

NONLINEAR ANALYSIS OF AIRFOIL FLUTTER AT TRANSONIC SPEEDS

I. Introduction

Aeroelasticity has been an area of research interest for many years. This interest is due to the catastrophic effects of the aeroelastic phenomenon called *flutter*. Flutter is an oscillatory aerodynamic condition resulting from fluid-structure interaction. The instability can be associated with the formation of large vortical structures, as in the case of the low-speed, high angle-of-attack flow regime or associated with complex shock wave motion in the transonic regime. The high-frequency and large-amplitude motion that results from flutter can cause loss of structural integrity in wings. For this reason, the prediction of flutter is of the utmost importance.

Researchers have typically computed the onset of flutter through simulations of the time-accurate behavior of the aeroelastic body [40]. The flutter onset point is located by bracketing the stability transition point with multiple time-integration solutions. This method can be computationally expensive for solutions of the Navier-Stokes equations for two reasons. First, each simulation requires an extended total time to simulate the time-periodic behavior near the flutter point without transients. Second, several simulations are necessary to bracket the stability transition point.

An alternate method of computing a flutter point is to directly compute the stability transition point using an extended system of equations comprised of the equilibrium equations and conditions related to flutter onset. Direct computation of the flutter onset point is the focus of the current research.

1.1 Historical Perspective on Flutter

One of the earliest analytical treatments of flutter was published by Theodorsen [129]. He showed that a two-degree-of-freedom airfoil model could achieve a condition of instability. His analysis is based on the assumption of potential flow. In a follow-on study by Theodorsen and Garrick [130], a rectangular wing section was tested in a wind tunnel at 202 mph and the violent flutter motion was photographed (Figure 1.1). Since then, many experiments, analytical studies, and computational studies have been performed for incompressible and compressible flow.

Edwards and Thomas [40] and Edwards and Malone [41] have written summary papers on the transonic aeroelasticity problem. In these papers they discuss the various types of oscillatory flow, the various mathematical models necessary to simulate the phenomenon, and a list of experimental data to be used for comparison. Reference [40] primarily discusses two-dimensional modeling and experiments, whereas reference [41] considers the three-dimensional problem.

1.1.1 Summary of Experimental Data. The unsteady airfoil data available prior to 1987 was reviewed by Edwards and Thomas [40] and is summarized in Table 1.1. Table 1.1 is a cross-referencing of airfoil type, experimental data, and numerical method. Every reference in Table 1.1 has experimental data for the airfoil listed. Headings for the columns denote various governing equations solved by the numerical methods. As one can see from Table 1.1, many different airfoils and numerical methods have been used in the past. The first three airfoils are conventional airfoils with a variation in the thickness ratio of 6, 10, and 12 percent. Tijdeman [133] tested the NACA 64A006 airfoil with a trailing-edge control surface allowed to oscillate about its quarter chord. These tests have become a classic source for the classification of transonic flow phenomena. Davis and Malcolm [32] tested the NACA 64A010A airfoil for pitching oscillations. Two of the cases from these tests have become widely studied: a moderate shock wave case at a freestream Mach number, M_∞, of 0.8 and an angle-of-attack, α, of 0°, and a case with steady shock induced separation at $M_\infty = 0.8$ and $\alpha = 4^o$. Landon [10] performed a study of the NACA 0012 airfoil in which larger

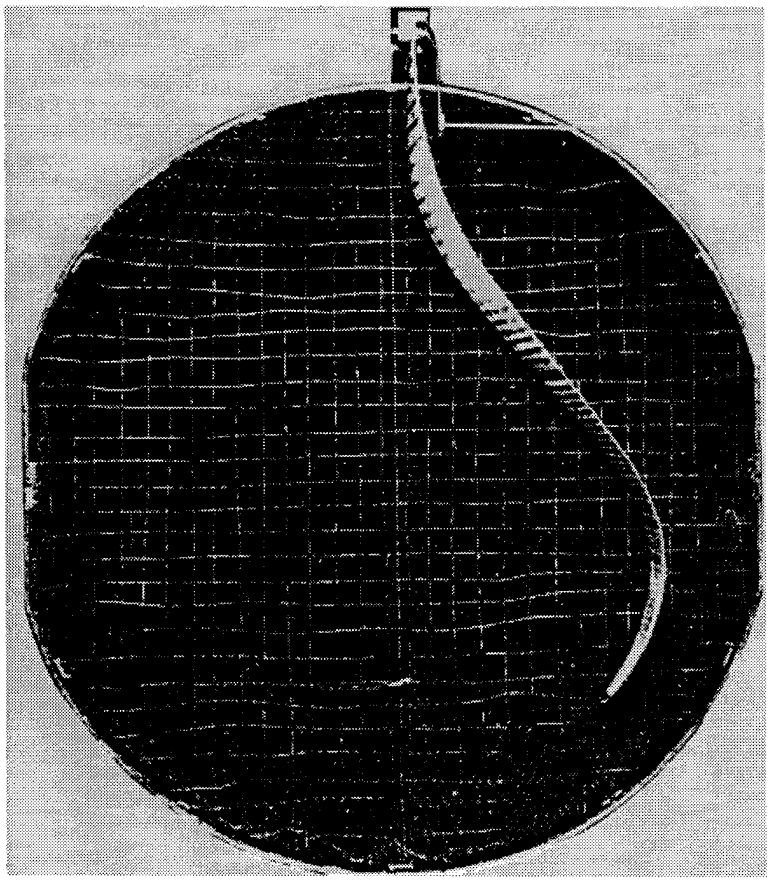


Figure 1.1 The Effect of Violent Flutter, Theodorsen and Garrick [130]

dynamic pitching amplitude motion and transient ramping motion was conducted, making the data suitable for dynamic stall computational studies. McDevitt and Okino [91] reported measurements of periodic shock induced oscillations for the NACA 0012 airfoil.

References for several supercritical airfoil studies are also provided in Table 1.1. Tijdeman and Davis [76] provide data for the 16 percent thick supercritical NLR 7301 airfoil including the shock free condition. Zimmerman [152] presents data for the 8.9 percent thick MBB A-3 airfoil, also including the shock free condition. Other supercritical airfoils tested for oscillatory motion or unsteady behavior are a 12 percent thick airfoil tested for pitching, heaving, and flap rotation by den Boer and Houwink [33], the RA16SC1 airfoil tested by ONERA [62], and a cryogenic test of a supercritical SC(2)-0714 airfoil by Hess et al. [51]. Reference [33] reported large dynamic responses of airloads on the supercritical airfoil for both oscillating and static motions for mixed separated and attached flow conditions.

Circular-arc airfoils have also served as the basis for the establishment of standard comparison data sets. References [89] and [90] provide detailed results for an 18 percent thick airfoil for Reynolds numbers ranging from 1 million to 17 million, covering laminar to fully developed turbulent flows. Time-periodic flows were observed over a narrow range of Mach numbers, which differed depending on whether the Mach number was increasing or decreasing. Mabey [80] studied similar periodic flows for a series of circular arc airfoils ranging in thickness between 10 and 20 percent over Reynolds numbers between 0.4 and 0.6 million. Reference [81] further investigates a larger, 14 percent thick circular-arc airfoil.

Rivera et al. [103, 104, 105, 12, 106] from NASA Langley have been involved in the "Benchmark Aeroelastic Models Program", which has the primary purpose of providing the necessary data to evaluate computational fluid dynamics codes for aeroelastic analysis. This series of experimental tests utilizes a rigid, rectangular planform model. The model is allowed to pitch and plunge and the angle-of-attack for zero structural load can be varied. Three different airfoil sections are being

Airfoil	TSD	FP	EE	NS
NACA 64A006	[5],[114]*,[53],[141]*	[116],[46]	[82],[83]	
NACA 64A010A	[57],[28],[47],[53], [38],[107]*,[141],[39], [143]+,[86]*,[29]*-[42]*	[85],[46], [52],[48]	[83],[118],[120]	[27],[26]
NACA 0012	[53],[19]*,[65]*,[143]+, [127]+,[39]	[48],[99]+	[60],[9],[59]	[116],[3],[21], [66],[54]
NLR 7301	[38],[53],[64]*, [86]*,[143]+	[85]	[84],[118],[120]	
MBB A-3	[107]*,[38],[28], [47],[86]*,[91]			
Supercritical	[54]*,[65]*,[33]*,[62]			
Circular Arc	[63]*,[64]*,[42]*			[89],[67],[90],[68]
Other			[54]	[125],[100],[66], [21]

Table 1.1 Index of References for Experimental and Numerical Data extracted from [40] (*-interacted boundary layer model, +-nonisentropic corrections, TSD-transonic small disturbance equation, FP-full potential equation, EE-Euler equations, NS-Navier-Stokes equations)

tested: the NACA 0012 airfoil, the NASA SC(2)-0414, and the NACA 64A010 airfoil. The model is instrumented at two spanwise locations (60% and 95%) for pressure, and accelerometers are placed on the wing planform as well. Other configurations include a trailing-edge control surface which can be actively controlled. Reference [106] includes plots of stability boundaries for both variations in Mach number and angle-of-attack. Steady and unsteady pressure distributions are also provided for a pinned and unpinned model, respectively.

1.1.2 Summary of Numerical Methods. The choice of method for the numerical simulation of flutter is highly dependent on the flow conditions that exist. Edwards and Thomas [40] and Edwards and Malone [41] discuss three flow types. Type 1 flows include one of the most important aeroelastic analysis conditions, cruise at high dynamic pressure (attached flows). Type 2 flows represent an increase in speed or angle-of-attack from the type 1 conditions resulting in a mixed attached and separated flow region. The mixed flow can result in significant unsteadiness even for rigid airfoils. Type 3 flows result from even further increases in speed or angle-of-attack and are characterized by fully separated flow regions.

1.1.2.1 Time-Integration Methods. As Table 1.1 depicts, there are a variety of computational methods available to analyze the aeroelastic phenomenon of flutter. The majority of the analyses have been accomplished with various forms of the potential-flow equation and have been limited to type 1 flow conditions. Both TSD and FP formulations have been used, but due to the lack of boundary-layer modeling, type 2 and 3 flows could not be simulated. These methods sometimes use a linearization of the airfoil surface boundary conditions by applying them at a mean airfoil surface position, thereby allowing a fixed grid to be used. Extensions of the FP and TSD codes include interactive boundary-layer models, which solve the boundary-layer equations after a potential flow solution has been found, and then iterate this procedure to a solution. Also, some researchers have developed corrections to the potential flow equations to include boundary-layer effects. References for the FP and TSD equations are given in Table 1.1.

The majority of modern flutter simulation is with the unsteady Euler equations. Euler methods primarily incorporate higher-order schemes to capture accurately the complex shock wave motions associated with transonic flows. The improved Euler codes tend to enforce accurately the airfoil surface boundary conditions and either lag the dynamic airfoil motion or solve the Euler equations and the dynamic motion of the airfoil simultaneously. References for solutions to the Euler equations are also given in Table 1.1. Note that the number of references are much less than TSD or FP, but still include supercritical airfoil calculations.

Bendiksen and Kousen [10] developed a time-integration scheme to analyze transonic flutter of an airfoil with pitch and plunge freedom. The unsteady airloads were computed by solving the Euler equations with a finite-volume scheme on a moving mesh. The structural model was comprised of linear and torsional springs. Two second-order differential equations govern the motion in pitch and plunge. The authors computed solutions for two cases of a NACA 64A010 and one case of a NACA 64A006 studied by other researchers. The flutter boundaries for these cases are obtained by bracketing the boundaries with time-integration solutions. Solutions were obtained by first

oscillating the airfoil in pitch for 3-6 cycles at an amplitude of 0.1° and then releasing the airfoil. The resulting oscillatory behavior was used to define the stability of the system for the particular parameters chosen. By varying the reduced velocity and holding other parameters constant, a plot of the steady-state angle-of-attack and maximum amplitude of the oscillatory angle-of-attack was generated and the flutter onset point documented. They state that the results are in overall good agreement with other researchers results except where strong shocks are present.

Kousen and Bendiksen [73] use the previously described method [10] with the modification of allowing a nonzero static pretwist. The freedom allowed oscillations about nonzero angles-of-attack. The computed Hopf-bifurcation curves for Mach numbers in the range $0.85 - 0.92$ were provided. They discovered a weak divergence for a small range of reduced velocity. This weak divergence was shown to be an attractor in neighboring limit-cycle solutions.

Kousen and Bendiksen [74] further modified the code to allow a nonlinear structural model. The structural nonlinearity was a structural freeplay, whereby the torsional moment goes to zero for a range of angles-of-attack symmetric about zero. The static freeplay was shown to significantly decrease the critical reduced velocity. Bendiksen [11] also modified the code to allow a moving trailing-edge flap. The computed solutions show a nonclassical form of shock-induced aileron buzz for certain structural parameters. Bendiksen states that the results suggest that shock-induced separation may not be an essential driving force for all "buzz" phenomena.

A few references listed in Table 1.1 model the fully viscous, shock-boundary-layer interaction problem by the Navier-Stokes equations with a turbulence model. These methods employ either explicit or implicit time-integration, with the majority using higher-order methods such as TVD schemes.

A representative study of viscous flutter is by Wu, Kaza and Sankar [146]. They developed a method of analyzing both transonic flutter and stall flutter. Their method solved the compressible Navier-Stokes equations in a body fitted, moving coordinate system using an approximate factor-

ization scheme. The structural equations were for pitch and plunge motion and were integrated in time with an Euler implicit scheme. They computed a viscous solution for a NACA 0012 undergoing dynamic stall at a freestream Mach number of 0.283 and a Reynolds number of 3.5 million. They also computed an Euler solution of a NACA 64A010 undergoing forced oscillatory motion at a freestream Mach number of 0.8. Both solutions were computed to validate the method against experiments and published numerical solutions. They also computed a flutter boundary as a function of the airfoil mass ratio using the Euler method for a NACA 64A006 airfoil at a freestream Mach number of 0.85. Using the same Mach number and structural parameters as used in the flutter boundary of inviscid solutions, a viscous solution was computed for a Reynolds number of 9 million. The airfoil mass ratio for flutter was slightly lower for the viscous solution ($\mu_s = 170$ for the viscous solution versus $\mu_s = 174.75$ for an inviscid solution) with a consistently higher amplitude of motion. They made the statement that the viscous and inviscid flutter boundaries were within 2% of each other, which reinforces their belief that in studies of high Reynolds number transonic flutter, inviscid calculations would suffice. They conclude that more analysis should be performed to study the effects of finer grid resolutions and different turbulence models.

A time-integration work related to the current research is by Buxton [22]. Buxton compared two shock capturing methods applied to transonic airfoil flutter. One of the methods compared was the time-integration method developed for the current research by the author. The other method is a 3-D Beam-Warming, Navier-Stokes/Euler solver, ENS3DAE [121]. Buxton modified ENS3DAE to allow pitch and plunge freedom of a rigid, rectangular wing by adding the structural model developed in the current research (discussed in later chapters in detail). Two-dimensional flow was enforced by placing inviscid planes at the wing tips.

Buxton found that, although the shock capturing method of the current method produced crisper shocks, the flutter onset point compared within 1%. The methods did show larger differences for the amplitude of oscillation of stable limit-cycles. ENS3DAE solutions were more dissipative

than the solutions of the current method, in general. (Comparisons from this work are provided in greater detail in Chapter V.)

1.1.2.2 Hopf-Bifurcation Methods. An alternate method of computing transitions from stable steady-state solutions to oscillatory solutions (flutter) is with bifurcation theory. Chapman and Tobak [25] review the forms of bifurcation which can occur between steady and unsteady flows. One of the aerodynamic problems discussed by Chapman and Tobak [25] involves a Hopf-bifurcation, due to transonic shock wave motion coupled with a dynamical system.

Hopf-bifurcation mathematically describes the transition point from a steady-state solution to an oscillatory solution [115]. Hopf-bifurcation points are characterized by having a complex pair of eigenvalues of the system Jacobian matrix with zero real part. Also, local to the Hopf-point, the real part of the eigenvalue pair must transition from negative to positive as a system parameter is varied.

Hui and Tobak [55] applied Hopf-bifurcation theory to the computation of the onset of pitching motion for a flat plate in a supersonic or hypersonic freestream. The aerodynamic model was limited to an assumed slowly varying harmonic motion. The freestream Mach number was assumed to be supersonic or hypersonic, omitting transonic effects. The dynamic system was simplified to a flat plate with pitch only freedom. Solutions were computed for a variety of supersonic Mach numbers.

A very general bifurcation analysis was developed by Griewank and Reddien [45]. They appended conditions on the eigenvalues of the fully implicit Jacobian matrix to the chosen governing equations, allowing a direct computation of the Hopf-bifurcation point. Their algorithm was applied to a one-dimensional convection-diffusion-reaction model problem. Several bifurcation points were computed for various system parameters, demonstrating the viability of the algorithm.

Jackson [58] provided the earliest application of the GR algorithm to two-dimensional flow past a rigid body modeled with the Navier-Stokes equations. He applied the Navier-Stokes equations with the assumptions of viscous, laminar, incompressible flow to rigid cylinders, ellipses, flat plates,

and wedges. The governing equations were discretized with a standard Galerkin finite-element method. The algorithm was developed to compute directly the stability transition point from stable, steady-state flow, to time-periodic flow for the various body shapes. Jackson cast the nondimensional governing equations in such a way as to have the ratio of major and minor axes as a parameter of the system. The angle of rotation between the minor axis and the freestream velocity vector was also a parameter of the system.

Jackson presented the idea of a consistency check of the Hopf-point with independently computed eigenvalues. He used an inverse iteration method of computing the generalized eigenvalues and eigenvectors of a nonsymmetric Jacobian matrix. The eigenvalue analysis provided both an initial guess of the eigenvector for the first Hopf-point calculation and a consistency check with the directly computed Hopf-point.

Jackson points out that the Hopf-point calculation converged only from "good" initial guesses for the basic flow, the critical Reynolds number (Hopf-point parameter), the critical Strouhal number (critical eigenvalue), and the eigenvector of the bifurcating solution. The procedure to compute a solution of a circular cylinder was to first use experimental data to specify a critical Reynolds number. Next, the complex eigenvalue nearest to $\pm 0.1\mathbf{i}$ and its eigenvector related to this Reynolds number were computed with eigenvalue analysis. Using this initial data set, solutions generally converged rapidly to the Hopf-point. For a given grid, additional Hopf-points for changes in geometric and aerodynamic parameters were computed with initial guesses based on neighboring Hopf-points.

Jackson computed solutions for ratios between 10^{-4} (representing a flat plate) and 2 (an ellipse). Solutions for increasing rotation angle produced a corresponding computational grid with increasing distortion to allow previous solutions to be used as initial guesses. The undistorted circular cylinder solutions compared very well with experimental data and time-integration solutions. Rotated ellipses, and wedges of different apex angles were also computed, demonstrating

the viability of the GR algorithm to compute solutions for a wide variety of shapes. Jackson notes that a companion time-integration method would allow flowfield solutions past the Hopf-point to be analyzed.

The works of Lutton [78], Beran and Lutton [15], and Lutton and Beran [79], applied the GR algorithm to a static NACA 0012 airfoil at various angles-of-attack. In references [78] and [15] a suite of time-integration, eigenvalue analysis, and Hopf-bifurcation algorithms were developed to analyze the point at which the flow over an airfoil becomes time-periodic for large angles-of-attack.

The streamfunction-vorticity form of the Navier-Stokes equations was used to model viscous, laminar, incompressible flow over an airfoil. A lack of time-dependent terms in the streamfunction equation caused a very complex form of the eigenvalue problem. The streamfunction-vorticity form of the eigenvalue problem involved a full, nonsymmetric matrix of rank equal to the number of grid points. The full matrix eigenvalue problem affected both the eigenvalue analysis and the GR algorithm applied to the streamfunction-vorticity equations. The details of how the GR algorithm was affected is discussed in Chapter VI.

Beran and Lutton [15] computed the critical Reynolds number for the onset of periodic motion as a function of angle-of-attack. The computed stability boundaries were validated with time-integration and eigenvalue analysis. Typically, only four iterations were required to compute an onset point. The stability boundary computed was the first directly computed flutter boundary for viscous flow over an airfoil modeled with the Navier-Stokes equations, as determined by an extensive computer search of the Aerospace Database. Unfortunately, the GR algorithm applied to the stream function-vorticity equations required $O(N^3)$ operations, where N represents the number of grid points. The authors state that the operation count practically limited the grids to $N \leq 4000$ for their analysis.

The time-integration method was extended to allow pitch and plunge motion [78, 79]. The structural equations governing the pitch and plunge motion were comprised of linear and torsional

springs with structural damping in both axes, and a nonzero unloaded angle-of-attack or static pretwist. The structural equations were integrated forward in time with a fourth-order Runge-Kutta algorithm loosely coupled with the aerodynamics solver. Flutter boundaries were computed for many structural and aerodynamic parameters. Also, the location of the Hopf-point for an airfoil with structural coupling was computed with the time-integration method and compared with the Hopf-point for a static airfoil computed with the direct method.

Lutton [79] describes attempts at extending the GR algorithm applied to the streamfunction-vorticity equations to allow pitch and plunge structural motion. Extending the direct method to allow pitch and plunge freedom was not straightforward and was abandoned after limited attempts due to changing priorities in the research effort.

The works of Lutton [78], Beran and Lutton [15], and Lutton and Beran [79] formed a foundation for the current research. Their successes with the GR algorithm applied to the incompressible Navier-Stokes equations aided in developing the necessary functional analysis tools to compute transonic flutter boundaries. Also, Lutton's [78] development of the pitch and plunge structural model and fourth-order Runge-Kutta time-integration scheme provided the necessary structural dynamics foundation for the current research and ongoing research by other investigators (Smith [119] and Buxton [22]).

The structural dynamics equations of Lutton [78] were nondimensionalized, in a form compatible with structural parameters in the literature and the aerodynamics equations, by Smith [119] and the author. Smith's research extends the work of Lutton and Beran [79] to include a trailing-edge control surface to delay the onset of flutter.

To demonstrate the validity of coupling the developed structural dynamics equations with aerodynamic equations suitable for computing flows with shocks, the eigenvalue analysis of Jackson [58] and Beran and Lutton [15] was applied to a pitch and plunge airfoil at transonic Mach number conditions by Morton and Beran [94]. The method was further extended to apply a modified

form of the GR algorithm to the structural coupling model by Morton and Beran [95]. The details of these two references are contained in this document as well as new and important developments.

During the course of computing comparison time-integration solutions, it became apparent how computationally expensive time-integration methods are [94, 95]. In general, accurate computation of a flutter point involved four or more oscillatory solutions, with two of them close to the stability point. Due to the light damping, solutions near the flutter point required up to 10^6 explicit iterations.

1.1.2.3 Eigen-Mode Time-Integration Method. Hopf-bifurcation analysis is based on a transition of eigenvalues of the fully implicit Jacobian matrix of a system of governing equations [115]. This transition is important since complex eigenvalues represent oscillatory modes of the system. References by Dowell [35] and Romanowski and Dowell [109, 110] develop a method of using the eigenvalues to approximate efficiently the time-accurate behavior of a fluid-structure interaction system. The method is a combination of the foundation mathematics in Hopf-bifurcation analysis and time-integration methods, and can be summarized in the following way:

- a steady-state solution is computed with a Beam-Warming ADI Euler or Navier-Stokes time-integration method,
- the Jacobian matrix is formed analytically with the solution computed in the first step,
- a reduced order set of eigen-modes are computed with an efficient eigenvalue-eigenvector algorithm,
- the eigen-modes are used to form a reduced order aerodynamic model,
- the reduced order aerodynamic model is coupled with the structural modes to simulate a fluid-structure interaction system.

They found that a very small number of modes is adequate in simulating the dynamic behavior and resulted in vast improvements in efficiency for a coarse grid over standard time-integration

methods. As the grid was refined, the number of necessary eigen-modes increased but was still far fewer than the number of degrees of freedom ($< 1\%$). All of the simulations reviewed are for Mach numbers less than 0.5. Their method is included in this review because they utilize some of the same foundational mathematics as the Hopf-bifurcation methods and because the method shows promise as a tool to analyze time-oscillatory solutions near the directly computed Hopf-bifurcation point. It is unknown whether the method is suitable for unsteady flows with strong shocks or for viscous, separated flows.

A significant contribution of the work of Romanowski and Dowell is their development of a very efficient algorithm to compute a reduced set of eigenvalues and corresponding eigenvectors of large nonsymmetric matrices, typically encountered with the discretized Euler equations, termed the Modified WYD algorithm. The resulting method was demonstrated to compute a reduced set of eigenvalues and eigenvectors for systems with on the order of 10^4 degrees of freedom.

A point of concern is the method of obtaining the equilibrium solution to linearize about. Romanowski and Dowell obtain solutions to linearize about using a time-integration method, from which a reduced order model is computed [109]. Since time-integration methods provide time-oscillatory solutions for conditions past stability boundaries, it is unclear as to how this method in present form can simulate stable limit-cycles when the unstable mode is an aerodynamic mode. Also, stable solutions very close to the stability boundary are computationally expensive to obtain due to the light damping, the very situation the current research is intentionally avoiding with an equilibrium method. It should be noted however, that the steady-state solution necessary in their method could be computed with equilibrium methods such as the one developed in the current research.

1.1.3 Nonlinear Structural Models. It has been shown by many investigators that for some regions of the flutter envelope, linear aerodynamic models are not adequate to model the flutter boundary of a physical system [40, 41]. Several investigators have improved the aerodynamic model

by implementing the nonlinear Euler or Navier-Stokes equations [40, 95]. Still, for many flutter cases, there remains disagreement between analysis and test results attributable to concentrated structural nonlinearities [20]. To improve simulations of fluid-structure interactions near flutter boundaries, nonlinear structural models must be used [20].

Reference [20] discusses several different types of concentrated nonlinear structural models and presents a method of analyzing combinations of them for a linear aerodynamic model. Reference [134] presents a method of analyzing the nonlinear structural components of preload and freeplay with the transonic small disturbance aerodynamics model and analyzes the stability of the systems resulting. Reference [74] extends the time-integration limit cycle analysis based on the Euler equations with a linear structural model to a nonlinear freeplay model. The results of this reference show significant movement of the flutter boundary for a single freeplay magnitude. Also, re-stabilization is presented for a NACA 64A010 with an elastic axis at the leading edge. The associated Hopf-bifurcation curve displays three Hopf-points.

Structural nonlinearities add a level of complexity that is not prohibited by the Hopf-point algorithm discussed previously. The change in stability behavior is so dramatic that designers of aeroelastic systems are interested in tools to analyze this class of structural models. For these reasons the current research is applied to the structural nonlinearity problem.

1.2 Research Scope and Objectives

The purpose of the current research is to improve the efficiency with which flutter boundaries are computed for fluid-structure interaction systems capable of modeling nonlinear aerodynamics and structural dynamics. Since time-integration methods are computationally expensive for computing time-accurate solutions near stability boundaries and they also require two or more solutions to bracket a stability point, more direct methods are desirable. Hopf-bifurcation meth-

ods show promise for computing stability boundaries but have not been applied to fluid-structure interaction systems based on Euler or Navier-Stokes aerodynamic models.

The scope of this research is threefold:

- develop a high-level algorithm to compute the stability transition point of a dynamic system,
- maintain a general formulation which involves Euler or Navier-Stokes aerodynamic equations and a general description of the structural degrees of freedom,
- apply the approach to a two-dimensional airfoil with pitch and plunge freedom in the transonic flow regime to demonstrate the gains in efficiency.

The following two subsections describe specific objectives of the research and a summary of the research method. The last subsection outlines the research document.

1.2.1 Research Objectives. There are seven specific research objectives:

1. Develop a 2-D, inviscid, shock-capturing, computer code to compute time-integration and equilibrium solutions for a static airfoil while
   - ensuring the time-integration method is completely consistent with the equilibrium method,
   - ensuring the equilibrium method is easily extensible to viscous flow problems and fluid-structure interaction problems.
2. Extend the time-integration code to admit two degrees of structural motion (pitch and plunge) for the rigid airfoil.
3. Add a coupled fluid-structure interaction model with both linear and bilinear torsional moment model for pitch and plunge freedom (PAPA) to the time-integration and equilibrium algorithms.
4. Validate the code for three classes of solutions

- static flowfield solutions,
- prescribed dynamic motion solutions,
- PAPA solutions (equilibrium and time-integration).

5. Develop an algorithm to directly compute the stability transition point of a fluid-structure interaction system with two criteria:
   - a workload on the order of a regular point calculation per iteration,
   - no fundamental restrictions from extending the method to a three-dimensional problem, (assuming infinite computational resources are available).
6. Apply the algorithm to a PAPA model and validate the solution with time-integration and eigenvalue analysis.
7. Compute flutter boundaries for various aerodynamic and structural parameters.

1.2.2 Research Solution Method. To model the flutter of airfoils at transonic speeds, several complex flow features must be addressed: unsteady motion, strong shock waves, and moving boundaries. The dynamic motion of the airfoil is modeled through a two-degree-of-freedom pitch and plunge model consisting of linear and torsional springs [129]. This model has been extensively used by many of the studies previously discussed and provides the freedom for complex, coupled, rotational and translational motion. A total variation diminishing (TVD) scheme is used to solve the Euler equations in strong-conservation form [149]. This approach accurately models shock wave structures at transonic Mach numbers [149] and is representative of modern flow solvers. The system of nonlinear, algebraic equations resulting from discretization of the governing equations is solved by Newton's method. Hopf-bifurcation points, which signal flutter onset, are directly computed on paths of equilibrium solutions following the procedure of Beran and Lutton [15].

Several of the tools necessary to successfully meet the objectives of the research have been used for other fluid dynamic problems. The four main tools necessary are Newton's method of computing

stable and unstable equilibrium solutions, Jacobian matrix elements computed numerically, linear system solvers for bordered, banded matrices, and iterative forms of the Hopf algorithm. The following subsections give representative works which utilize three of these methods. Although each of the first three tools have been used before, an algorithm to compute flutter boundaries incorporating all of them is unique.

1.2.2.1 Newton's Method. Newton's method is an attractive method to compute stable and unstable equilibrium solutions because of its robustness and quadratic convergence. The method has been applied to many engineering problems. This section describes a few of the applications of Newton's method to Euler or Navier-Stokes equations.

In the area of vortex breakdown, several authors have applied Newton's method. Beran [13] applied Newton's method to axisymmetric, incompressible, vortex breakdown. Due to the rapid convergence of Newton's method, Beran was able to compute a wealth of solutions for variations in Reynolds number, vortex strength, and other parameters of the system, including nonunique solutions. Morton [96] extended Beran's method to include compressibility effects on the vortex breakdown problem. Tromp [135] used Newton's method to efficiently compute axisymmetric solutions which he used as initial conditions for a 3-D, time-accurate, Beam-Warming algorithm. His analysis described the transition from 2-D vortex breakdown to 3-D vortex breakdown in a pipe.

Newton's method has also been applied to flows with shocks. Hafez, Palaswamy, and Mariani [49] applied Newton's method to a discrete, upwind and central difference form of the Euler equations and computed transonic flows over a cylinder and a NACA 0012 airfoil. In all cases, quadratic convergence rates were found. Venkatakrishnan [136] applied Newton' method to inviscid and laminar viscous solutions of a NACA 0012 airfoil at transonic Mach number conditions. He explored methods of accelerating convergence and increasing the efficiency of the linear system solvers.

1.2.2.2 Numerical Jacobians. An important element in Newton's method is the system Jacobian matrix. Complex shock capturing methods of solving the Euler and Navier-Stokes equations make the error free analytical determination of Jacobian matrices extremely difficult in a timely manner. Some researchers have computed the Jacobian matrices numerically to circumvent this problem. An added benefit of Jacobian matrices which are computed numerically is the relative ease with which the algorithm can be modified. With analytical Jacobian elements, a change to the system of equations means a change in the computer code for each affected Jacobian element. On the other hand, using a numerical procedure to compute the Jacobians is independent of the specific nature of each equation in the system and is thus programmed once.

Orkwis [98] applied a method of obtaining the Jacobian matrix numerically to computing Navier-Stokes solutions of flows with shocks. His method incorporated a Roe flux-difference-splitting scheme to enhance shock capturing and includes viscous terms for laminar flow. Jacobian elements were computed with a first-order finite-difference approximation. The method was compared with analytical Jacobians to assess convergence trends. Orkwis has found that in all cases, Newton's method with numerical Jacobian matrices converged faster than analytical Jacobian matrices. He states that this may be due to an error in the analytical Jacobian, pointing out one of the reasons for using numerical Jacobian elements. He states that the simplicity with which the numerical Jacobian elements are computed and the reliability of the Jacobian elements make the method a very useful one.

1.2.2.3 Bordered Systems. Discrete systems of equations associated with the Navier-Stokes equations typically produce Jacobian matrices with a large banded region. When coupling the aerodynamic equations to a small number of structural equations, the matrix is typically comprised of a large banded system bordered by a small number of rows and columns. The resulting linear system can no longer be solved with a standard banded matrix, linear system solver. A method of solving this type of linear system has been used in other contexts. Beran [13] applied a

border technique to solve a linear system resulting from the continuation procedure of Keller [70]. The increased workload of the bordered system solve over a banded system solve was negligible for the one bordered equation considered in this study.

1.2.2.4 Proposed Iterative Hopf Algorithm. Iterative forms of the GR algorithm applied to large aerodynamic systems have not been attempted. *Proposals* of iterative methods to solve the Hopf problem have appeared in two sources. The first source, reference [97], proposed a Gauss-Seidel method of eliminating manipulations which produced a full matrix in the work of Beran and Lutton [15]. The second reference, [16], generalized the method, allowing Euler or Navier-Stokes equations to be used as the aerodynamic model.

1.2.3 Research Overview. This section presents an overview of the research document. Chapter II describes the overall scheme for obtaining directly a stability transition point of an aerodynamic system. The first section defines relevant terms and analysis tools used in the method. The second section develops the extended system of equations describing the stability transition point. The last section outlines Newton's method, the foundational method of solving the nonlinear systems of equations for equilibrium and stability transition points.

Chapter III describes the governing aerodynamic equations for a static airfoil at an angle-of-attack. The first section gives the governing equations of the fluid dynamics equations in strong conservation law, nondimensional form. The scales used in nondimensionalization are also provided in this section. The second section extends the governing equations to include a generalized coordinate transformation for complex geometries. The third section describes the explicit time-integration scheme to obtain time-accurate solutions of the governing equations. The fourth section describes the computational mapping of grids used in the study and provides a complete list of all the grids used in the research. The fifth section describes the set of boundary conditions for each boundary. The sixth section details the method of computing force and moment coefficients from

the solution vector. The last section describes the method of computing directly the equilibrium solutions for a static airfoil at an angle-of-attack.

Chapter IV presents static airfoil results computed with the methods developed in Chapter III. The first section is a validation of the computer code by grid sensitivity studies and comparison with a solution of the open literature. The second section describes convergence properties of the equilibrium and time-integration methods. The last section presents equilibrium solutions for two airfoils, the NACA 0012 and the NACA 64A006 for a range of angles-of-attack and freestream Mach numbers.

Chapter V presents the fluid-structure interaction system, describes changes to the equilibrium and time-integration methods to allow airfoil motion, and then presents results. The first section details the structural model which allows pitch and plunge freedom, the changes to the governing equations for moving meshes, and modifications to the time-integration and equilibrium algorithms. The second section presents validation of the modified code and the last section presents equilibrium solutions for a NACA 64A006 airfoil for various static pretwists and freestream Mach numbers.

Chapter VI is an important chapter because it develops in detail a new method of directly computing the stability transition point of a fluid-structure interaction system. The first section presents the stability transition point system applied to the fluid-structure interaction model developed in Chapter V. The second section describes the most common method in the literature to solve the extended system and also presents the limitations of this method when applied to fluid-structure interaction systems. The last section presents a new algorithm which addresses the limitations of the other algorithm.

Chapter VII presents results of computations using the new algorithm of Chapter VI applied to a NACA 64A006 pitch and plunge airfoil model. The first section presents validation of the algorithm of Chapter VII through time-integration, eigenvalue analysis, and grid sensitivity. The

second section presents convergence properties of the stability transition point algorithm for various parameters. The last section presents flutter point results for a variety of aerodynamic and structural parameters.

Chapter VIII extends the algorithm to include a class of nonlinear structural models. The first section presents the equations of motion for the modified structural model and the second section describes validation of the code. The last section presents results for various structural parameters.

Chapter IX summarizes the conclusions from the current research and provides recommendations for future research. The first section presents a summary and conclusions for each of the previous chapters. The last section presents recommendations for future research.

There are five appendices which support the main chapters. Appendix A provides a detailed description of the shock capturing scheme used in the current research. First, the one-dimensional scheme is presented along with results from a one-dimensional model problem. The model problem combines the shock capturing scheme with the equilibrium solution scheme to validate the equilibrium solution approach without any geometric complexities. Next the scheme is generalized to include two-dimensional geometries with generalized coordinates and moving meshes.

Appendix B provides the formulation of the numerically computed Jacobian elements used in the equilibrium scheme. Also, results of equilibrium solutions for a two-dimensional model problem are presented to show the utility and performance of numerically computed Jacobian elements.

Appendix C describes a method of solving linear systems with a "bordered" structure. The bordering algorithm is an efficient way of solving a system with a large banded portion and a small number of full rows and columns bordering the banded elements.

Appendix D provides a single location for all of the run summary tables throughout the main chapters. Also, an estimate of the total number of CPU hours used in the research is provided. Finally, profile listings of the BGSN5 and BGSN4 codes are provided for comparison.

Appendix E is a summary of the computer code TVDntiAE, developed in the current research and a library of routines used by TVDntiAE. Also, information concerning the archived data resulting from the cases run in this research is provided.

II. Functional Analysis of Flows with Shocks

Seydel [115] presents an excellent overview of the elements of functional analysis. Relevant terms and analysis are extracted from [115] and presented in Section 2.1. Next, a method of extending a nonlinear system of equations to represent a Hopf-bifurcation point is described in Section 2.2. Finally, a method to obtain solutions of nonlinear systems, Newton's method, is reviewed in Section 2.3.

2.1 Elements of Functional Analysis

Nonlinear phenomena associated with solutions to nonlinear equations such as the Euler equations can be observed geometrically by the use of plots of the *solution space*. The solution space plot typically consists of the parameter to be varied, $\tilde{\lambda}$ (e.g., Mach number), on the abscissa and some scalar measure of the solution, $[y]$ (e.g., the angle-of-attack of the body or the lift coefficient), on the ordinate. The solution space plot can lead to insight into changes in system behavior with a change in some parameter value.

Points that make up the solution paths can be classified as solutions to the time-dependent equations (e.g., *steady-state*, *time-periodic*, or *aperiodic*) or solutions to the time-independent equations, (e.g., *equilibrium solutions*). Solutions to the time-dependent equations,

$$Y_t = G(Y; \tilde{\lambda}), \tag{2.1}$$

for which $Y_t = 0$ in the limit as $t \to \infty$ are called steady-state solutions. Time-periodic solutions (or *limit-cycles* [115]), satisfy (2.1) and also

$$Y(t + T_p) = Y(t), \tag{2.2}$$

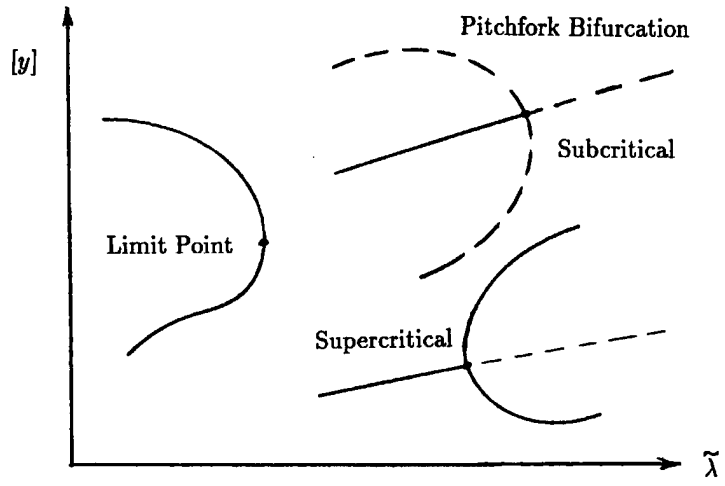


Figure 2.1 Examples of a Limit Point, and Sub- and Supercritical Pitchfork Bifurcation Points

for some period T_p. Aperiodic solutions are solutions of (2.1) that do not satisfy (2.2) for any T_p. An equilibrium solution can be described as a solution to the time-independent equations,

$$G(Y;\tilde{\lambda}) = 0. \tag{2.3}$$

Solution space plots may have multiple solution paths (*branches*) of either limit-cycle or equilibrium solutions. A bifurcation point is a crossing of two branches with distinct tangents. A *pitchfork bifurcation* (Figure 2.1) is an intersection of two branches, with one being one-sided.

There are two types of equilibrium solution points of a continuously differentiable system: *regular points* and *singular points.* A regular point is an equilibrium solution with a nonsingular Jacobian (i.e., $G(Y;\tilde{\lambda}) = 0$ and $G_Y \neq 0$), whereas a singular point $(Y_s, \tilde{\lambda}_s)$ has a singular Jacobian,

$$G_Y|_{(Y_s;\tilde{\lambda}_s)} = 0. \tag{2.4}$$

There are several types of singular points, including *limit points* and *singular bifurcation points.* A limit point (Figure 2.1) is a point at which there are no solutions to one side of the point and two

solutions on the other side, locally. A singular bifurcation point, on the other hand, is a point at which two branches of equilibrium solutions intersect with distinct tangents.

The solution points classified above are either stable or unstable to infinitesimal perturbations of the system. In general, steady-state solutions are stable, and equivalent to stable equilibrium solutions. Unstable equilibrium points may exist for which there are no corresponding steady-state solutions of the time-dependent equations.

A point of transition from stable solutions to unstable solutions (*stability transition point*) is of great interest to designers of systems. Singular points, such as limit points and singular bifurcation points, are in general stability transition points. *Supercritical bifurcations* have stable solutions on both sides of the bifurcation point, whereas, *subcritical bifurcations* have unstable solutions on both sides of the bifurcation point. Figure 2.1 depicts super- and subcritical pitchfork bifurcation points, where a solid line denotes stable solutions and a dashed line denotes unstable solution points.

A bifurcation point that does not satisfy (2.4) but can be a stability transition point is a *Hopf-bifurcation point* (Figure 2.2). A Hopf-bifurcation point connects an equilibrium solution path with a path of stable limit-cycle solutions. Hopf-bifurcation points may also be *bistable*, which denotes a change of stability on a single path (Figure 2.3). The periodic solutions are unstable (hollow dots in Figure 2.3) until a limit point is crossed at which point the solutions can become stable periodic solutions, solid dots in Figure 2.3.

There are various methods of computing stability transition points. One indirect method is based on computing a solution of (2.3) and comparing to a solution of (2.1) for $t \rightarrow \infty$. If the solutions are equivalent, the branch is considered stable. Through a variation in $\tilde{\lambda}$ the stability transition point can be bracketed. A direct method of computing a stability transition point is to expand (2.3) to include conditions that describe the stability transition point. Seydel [115] presents expanded systems which can be used to compute limit points and singular bifurcation

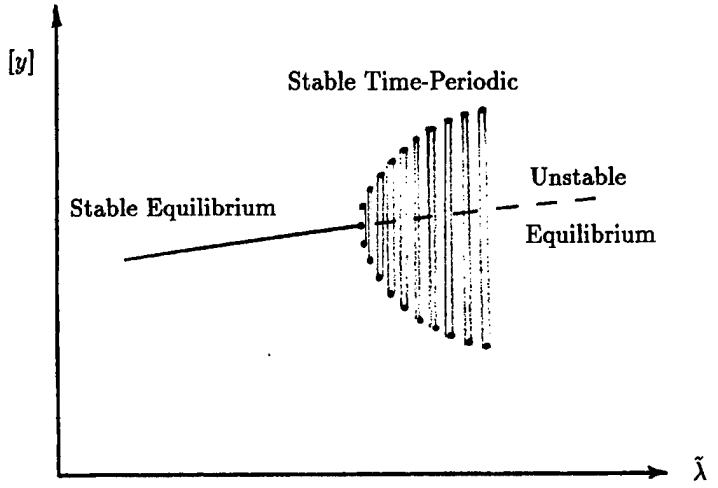


Figure 2.2 Solution Path With A Hopf-Point

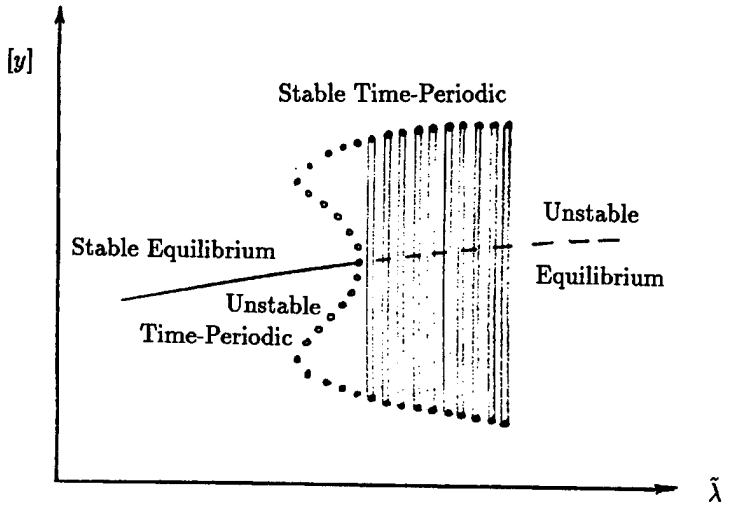


Figure 2.3 Solution Path With A Bistable Hopf-Point

points. Roose and Hlavacek [111] and Griewank and Reddien [45] develop expanded systems to compute a Hopf-bifurcation point. The expanded system of [45] is developed in the next section.

2.2 Hopf-Bifurcation System

At a Hopf-point, a steady-state solution transitions to a time-oscillatory solution with zero amplitude [115]. The Hopf-point can be expressed as an oscillatory perturbation to an equilibrium solution, Y^o:

$$Y = Y^o + \epsilon P e^{\beta t} + O(\epsilon^2), \tag{2.5}$$

where ϵ is a vanishingly small parameter, P is a coefficient vector, and β is a complex frequency. Using the assumed form of the solution, $G(Y;\tilde{\lambda})$ can be linearized about Y^o:

$$G(Y;\tilde{\lambda}) = G^o + \epsilon e^{\beta t} G^o_Y P + O(\epsilon^2), \tag{2.6}$$

where

$$G^o \equiv G(Y^o;\tilde{\lambda}), \quad G^o_Y \equiv G_Y(Y^o;\tilde{\lambda}). \tag{2.7}$$

The temporal derivative of (2.5) is

$$Y_t = \epsilon \beta P e^{\beta t} + O(\epsilon^2). \tag{2.8}$$

Substituting (2.8) and (2.6) into (2.1) yields

$$\epsilon \beta P e^{\beta t} - G^o - \epsilon e^{\beta t} G^o_Y P + O(\epsilon^2) = 0. \tag{2.9}$$

Grouping terms of like order gives

$$O(1): \qquad G^o = 0, \tag{2.10}$$

$$O(\epsilon): \qquad \epsilon\beta P e^{\beta t} - \epsilon e^{\beta t} G_Y^o P = 0. \tag{2.11}$$

Equation (2.10) is the familiar equilibrium problem, and (2.11) can be rewritten in the following form:

$$G_Y^o P = \beta P, \tag{2.12}$$

the classic eigen-problem. As seen in (2.12), the coefficient vector, P, is the eigenvector related to the eigenvalue β and in general has the complex form $P = P_1 + \mathbf{i}P_2$. By examining (2.5), it is observed that the eigenvalue with the largest real part, evaluated at the equilibrium solution, G^o, determines system stability [115]. If the real part of this eigenvalue is less than zero, the system is asymptotically stable. If the real part is greater than zero, the system is unstable.

Seydel [115] provides three conditions for a Hopf-point $(Y^*, \tilde{\lambda}^*)$ to exist:

- the time-independent equations are satisfied, $G(Y^*; \tilde{\lambda}^*) = 0$,
- the Jacobian matrix $G_Y^* \equiv G_Y(Y^*; \tilde{\lambda}^*)$ has a pair of purely imaginary eigenvalues with no other eigenvalue having vanishing real part, $\beta_\pm = \pm\mathbf{i}\Theta$ $(\Theta \neq 0)$,
- the "transversality condition" is satisfied: $\frac{d}{d\tilde{\lambda}} \left(Real \left[\beta(\tilde{\lambda}) \right] \right) \Big|_{\tilde{\lambda}=\tilde{\lambda}^*} \neq 0.$

If at a point $(Y^*, \tilde{\lambda}^*)$ the above three conditions hold, then from the point emerges a branch of solutions with limit-cycle behavior and with period $2\pi/\Theta$. Figure 2.2 shows a solution path with a Hopf-point. The solid dots past the Hopf-point in Figure 2.2 are peak values of the limit-cycle solutions. Using the second condition above, (2.12) can be separated into relationships for the real and imaginary components of the eigenvector:

$$G_Y P_1 + \Theta P_2 = 0, \tag{2.13}$$

$$G_Y P_2 - \Theta P_1 = 0. \tag{2.14}$$

Equations (2.10), (2.13) and (2.14) are three equations for five unknowns: Y, P_1, P_2, $\tilde{\lambda}$, and Θ. The remaining two relationships ensure the trivial solution is excluded [45]:

$$q^T P_1 = 0, \qquad q^T P_2 = 1, \tag{2.15}$$

where q is a constant normalization vector. In summary, a Hopf-point satisfies the nonlinear system of equations $\mathcal{F}(\mathcal{X}) = 0$, where

$$\mathcal{F} = \begin{bmatrix} G \\ G_Y P_1 + \Theta P_2 \\ G_Y P_2 - \Theta P_1 \\ q^T P_1 \\ q^T P_2 - 1 \end{bmatrix} = 0, \quad \mathcal{X} = \begin{bmatrix} Y \\ P_1 \\ P_2 \\ \tilde{\lambda} \\ \Theta \end{bmatrix}. \tag{2.16}$$

2.3 Newton's Method

Equilibrium solutions of the pitch and plunge airfoil model are computed with Newton's method in this investigation. Newton's method is an iterative, fully implicit method capable of computing both stable and unstable equilibrium solutions of the nonlinear system (2.3),

$$G(Y; \tilde{\lambda}) = 0. \tag{2.17}$$

The method is described as follows. Given Y^ν, an initial approximation to the solution vector Y which does not satisfy (2.17), a new approximation $Y^{\nu+1}$ is found for a specified value of $\tilde{\lambda}$ by solving the system of equations [151]

$$G_Y(Y^\nu; \tilde{\lambda}) \left(Y^{\nu+1} - Y^\nu\right) = -G(Y^\nu; \tilde{\lambda}). \tag{2.18}$$

One Newton iterate involves computing a solution to (2.18) and updating Y to find $Y^{\nu+1}$. Successive Newton iterates are computed until the L_2 norm of G,

$$\| G \|_2 \equiv \sqrt{\sum_{k=1}^{N} G_k^2(Y^{\nu+1}; \tilde{\lambda})}, \tag{2.19}$$

satisfies the condition

$$\| G \|_2 \leq \epsilon_{conv}, \tag{2.20}$$

where N is the number of unknowns of G and ϵ_{conv} is specified to be machine precision or some other tolerably small value. G_Y is the Jacobian matrix, which is defined as

$$G_Y \equiv \left[\frac{\partial G_i}{\partial Y_j}\right]. \tag{2.21}$$

Newton's method is an attractive algorithm because of its simplicity and its convergence rate. The method is guaranteed to converge *quadratically* if the Jacobian matrix is nonsingular and the initial guess is sufficiently close to the solution, i.e., within a *ball of convergence* [56] centered about the solution point. Quadratic convergence rate can be described as reducing the error twice the number of digits past the decimal for each successive iteration. Solutions to (2.17) are generally obtained in ten iterations or less.

III. Transonic Flows About Fixed Airfoils

To model accurately complex shock wave motion occurring over airfoils at transonic speeds, the Euler equations must be used [40]. Solutions to the Euler equations can be computed in a variety of ways, but a class of high-resolution upwind total-variation-diminishing (TVD) schemes has recently become very popular for this purpose. In particular, flows with discontinuities can be computed accurately by these TVD schemes. Harten [50] presents a class of explicit, second-order-accurate, highly nonlinear difference schemes. This chapter presents the governing equations and application to the two-dimensional, static airfoil at angle-of-attack. First, the governing equations to be solved are described. Next, the splitting form of the TVD numerical scheme is presented, leaving the details to Appendix A. Also, the computational grid and boundary conditions are described. Finally, a method of computing an equilibrium solution of the TVD numerical scheme is presented.

3.1 Governing Equations

The Euler equations are statements of the conservation laws for mass, momentum, and energy for an inviscid, non-heat conducting flow. When the Euler equations are arranged such that the *conserved variables* ρ, ρu, ρv, and E_t are dependent, the *conservative* form of the gasdynamic equations is obtained. The conservative form is necessary (not sufficient) when computing flows with discontinuities for the discontinuity to represent a physical wave when shock capturing schemes are applied. The method used to ensure sufficiency by eliminating nonphysical shocks is presented in Appendix A. The governing equations are written in the following vector form:

$$\frac{\partial \bar{U}}{\partial t} + \frac{\partial \bar{F}(\bar{U})}{\partial x} + \frac{\partial \bar{G}(\bar{U})}{\partial y} = 0, \tag{3.1}$$

where $\bar{U}$ contains the dependent variables ρ, ρu, ρv, and E_t. The term $\bar{F}$ contains the fluxes differentiated in (3.1) with respect to x, and the term $\bar{G}$ contains the fluxes differentiated in (3.1)

with respect to y. The elements of $\bar{U}$, $\bar{F}$, and $\bar{G}$ are

$$\bar{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E_t \end{bmatrix}, \quad \bar{F} = \begin{bmatrix} \rho u \\ (\rho u)^2/\rho + p \\ \rho uv \\ (E_t + p)\rho u/\rho \end{bmatrix}, \quad \bar{G} = \begin{bmatrix} \rho v \\ \rho vu \\ (\rho v)^2/\rho + p \\ (E_t + p)\rho v/\rho \end{bmatrix}. \tag{3.2}$$

The equations of state for a perfect gas with constant specific heats c_p and c_v are

$$p = (\gamma - 1)\rho e = (\gamma - 1)\left[E_t - \frac{\rho(u^2 + v^2)}{2}\right], \tag{3.3}$$

$$\gamma = \frac{c_p}{c_v}, \quad e = c_v T. \tag{3.4}$$

The primitive variables are nondimensionalized by the scalings

$$t^* = \frac{tV_\infty}{c}, \quad x^* = \frac{x}{c}, \quad y^* = \frac{y}{c}, \quad u^* = \frac{u}{V_\infty}, \quad v^* = \frac{v}{V_\infty}, \quad p^* = \frac{p}{\rho_\infty V_\infty^2}, \quad T^* = \frac{T}{T_\infty}, \quad e^* = \frac{e}{V_\infty^2}, \tag{3.5}$$

where c is the airfoil chord length, V_∞ is the magnitude of the freestream velocity vector, ρ_∞ is the freestream density, and T_∞ is the freestream temperature. The $*$ notation of this section denotes a nondimensional variable. These particular scalings allow the equation of state (3.3) to remain unchanged in nondimensional form and the internal energy-temperature relationship to become

$$e^* = \frac{T^*}{\gamma(\gamma - 1)M_\infty^2}. \tag{3.6}$$

After dropping the $*$ superscript from the non-dimensional quantities for notational simplicity, the governing equations and the definitions of $\bar{U}$, $\bar{F}$, and $\bar{G}$ in nondimensional form are equivalent to (3.1) and (3.2). Future references to (3.1) are considered to be the nondimensional form.

3.2 General Coordinate Transformation

Due to the complexity of the airfoil geometry, a general spatial transformation of the form $\xi = \xi(x,y)$ and $\eta = \eta(x,y)$ is used to transform (3.1) from the physical domain (x,y) to the computational domain (ξ,η) [150]. The advantage of this approach is the ability to use a body-conformal mapping, which makes implementation of boundary conditions much easier. Section 3.4 discusses the particular conformal mapping of points, or *grid*, to be used.

Through spatial transformation of the governing equations, (3.1), the following vector form is obtained:

$$\frac{\partial \hat{U}}{\partial t} + \frac{\partial \hat{F}(\hat{U})}{\partial \xi} + \frac{\partial \hat{G}(\hat{U})}{\partial \eta} = 0, \tag{3.7}$$

$$\hat{F}(\hat{U}) = (\xi_x \bar{F}(\bar{U}) + \xi_y \bar{G}(\bar{U}))/\mathcal{J}, \quad \hat{G}(\hat{U}) = (\eta_x \bar{F}(\bar{U}) + \eta_y \bar{G}(\bar{U}))/\mathcal{J}, \tag{3.8}$$

$$\hat{U} = \bar{U}/\mathcal{J}, \quad \mathcal{J} = \xi_x \eta_y - \xi_y \eta_x, \tag{3.9}$$

where $\mathcal{J}$ is the Jacobian of the transformation, and ξ_x, ξ_y, η_x, and η_y are the transformation metrics (subscript denotes differentiation with respect to the subscripted variable).

Figure 3.1 displays the relationship between the (ξ,η) coordinates and the (i,j) indices. By assumption

$$\Delta\xi = \Delta\eta = 1, \tag{3.10}$$

for a unit change in either coordinate direction index.

3.3 Time-Integration Scheme

Equation (3.7) is solved by employing an explicit algorithm that splits the multidimensional finite-difference algorithm into a sequence of one-dimensional operations [2]. The resulting algo-

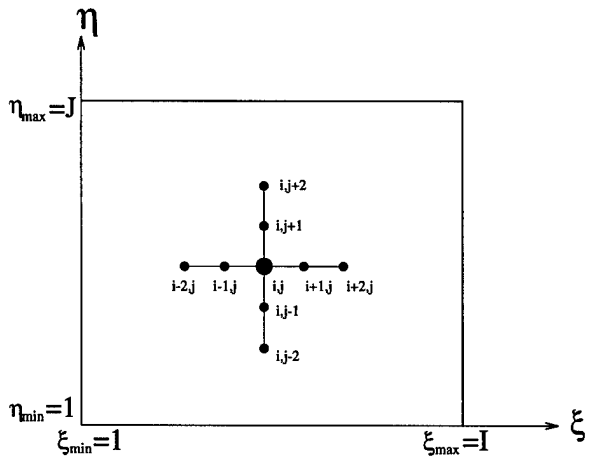


Figure 3.1 Computational Domain and Index Definition

rithm is described as

$$L_\eta \hat{U}^n_{i,j} = \hat{U}^{n+1/2}_{i,j} = \hat{U}^n_{i,j} - \Delta t \left(\tilde{\hat{G}}^n_{i,j+\frac{1}{2}} - \tilde{\hat{G}}^n_{i,j-\frac{1}{2}} \right), \tag{3.11}$$

$$L_\xi \hat{U}^{n+1/2}_{i,j} = \hat{U}^{n+1}_{i,j} = \hat{U}^{n+1/2}_{i,j} - \Delta t \left(\tilde{\hat{F}}^{n+1/2}_{i+\frac{1}{2},j} - \tilde{\hat{F}}^{n+1/2}_{i-\frac{1}{2},j} \right), \tag{3.12}$$

with

$$\hat{U}^{n+1}_{i,j} = L_\xi L_\eta \hat{U}^n_{i,j}. \tag{3.13}$$

Equation (3.13) defines an explicit iteration valid for

$$1 \leq i \leq I, \quad \text{and} \quad 2 \leq j \leq J-1. \tag{3.14}$$

The superscript $n+1/2$ denotes an intermediate stage in the time integration, n denotes the current time level, and $n+1$ denotes the current time level plus the time step, Δt. Equation (3.13) describes an $O(\Delta t,\ \Delta x^2,\ \Delta y^2)$ method of updating the flowfield variables. The definitions of $\tilde{\hat{F}}$ and $\tilde{\hat{G}}$ can be found in Appendix A. Figure 3.1 also presents the stencil of dependent variables necessary to

compute (3.11)-(3.13). The cases for which the computational stencil extends outside of the domain in the η direction are covered in Appendix A, and the ξ direction in Subsection 3.5.4.

3.4 Computational Grid

It is necessary to specify the discrete distribution of points over which the equations are solved. The chosen grid structure for this work is an "O"-grid topology. The trailing edge of the airfoil is replaced with a circular arc that matches the slope of the airfoil shape at some percentage of chord (i.e., 98% or 99%) to facilitate the "O"-grid topology. Figure 3.2 shows the relationship between the rounded trailing-edge airfoil and an airfoil with a sharp trailing edge. The length of the two airfoil types differs by Δte. The cut in the "O"-grid is placed along the chord line starting at the leading edge and ending at the outer edge of the domain. The grids are generated with GRIDGEN, an elliptic grid generator [126]. A circle of radius R_{max} with a uniform node distribution is specified as the outer boundary. The surface distribution is specified with GRIDGEN to provide clustering at high curvature. The inner and outer grid edges are connected with Vinokur progression [126] having an initial spacing at the airfoil surface of $\Delta wall$. The elliptic solver is used to enforce orthogonality at the domain edges. A representative grid for a NACA 64A006 airfoil is depicted in Figures 3.3 and 3.4. A discussion of the basis for choosing an "O"-grid over a "C"-grid for the current research is presented in Subsection 3.7.

Two airfoils are used throughout the research. The NACA 0012 is used because of the wealth of static airfoil data available in the open literature. It is a 12% thick, symmetric airfoil. The circular arc trailing edge is placed at 99.5% chord and results in an airfoil which is 99.55% chord in length. The NACA 64A006 airfoil is chosen because of the pitch and plunge time-accurate data available for comparison. It is a 6% thick symmetric airfoil. The circular arc trailing edge is placed at 98% chord due to the thickness of the airfoil and results in an airfoil 98.13% chord in length. Table 3.1 is a summary of the grids used throughout the research.

Grid#	*I*	*J*	R_{max}	Δ_{wall}
NACA 0012 Airfoil				
G12-1	80	120	100	0.001
G12-2	80	80	50	0.001
G12-3	80	64	25	0.001
G12-4	80	60	20	0.001
G12-5	80	56	15	0.001
G12-6	80	42	10	0.001
G12-7	80	42	10	0.003
G12-8	80	42	10	0.005
G12-9	80	32	10	0.005
G12-10	80	16	10	0.005
G12-11	160	32	10	0.005
G12-12	120	32	10	0.005
G12-13	80	32	10	0.005
G12-14	80	32	10	0.005
G12-15	80	32	10	0.005
G12-16	80	32	10	0.005
G12-17	80	32	10	0.005
G12-18	320	64	25	0.001
NACA 64A006 Airfoil				
G646-1	100	31	15	0.015
G646-2	100	15	10	0.001
G646-3	100	15	10	0.015
G646-4	100	31	15	0.005
G646-5	100	15	10	0.005
G646-6	60	15	8	0.015
G646-7	60	15	8	0.005
G646-8	60	15	10	0.005
G646-9	60	13	4.1	0.005
G646-10	30	9	6	0.005

Table 3.1 Grid Definitions

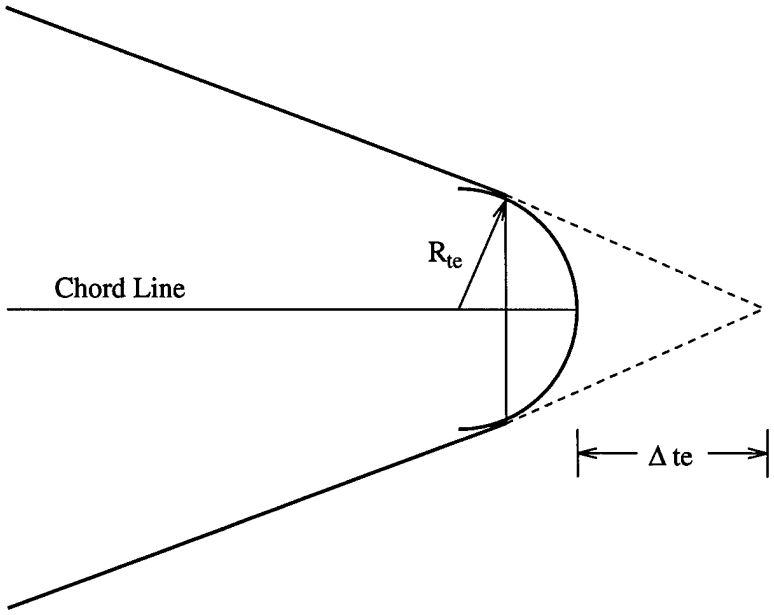


Figure 3.2 Trailing-Edge Schematic

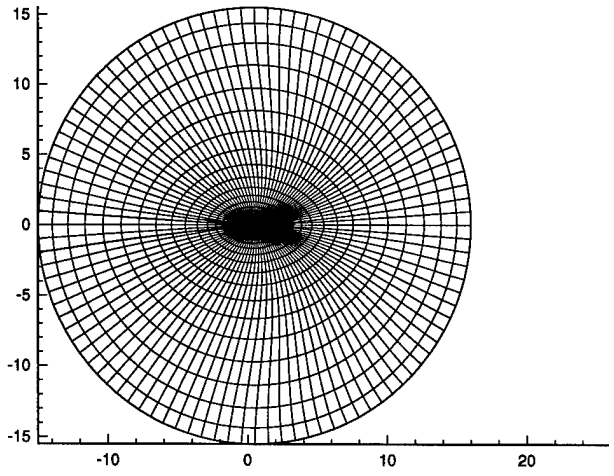


Figure 3.3 Representative Grid (NACA 64A006 Airfoil)

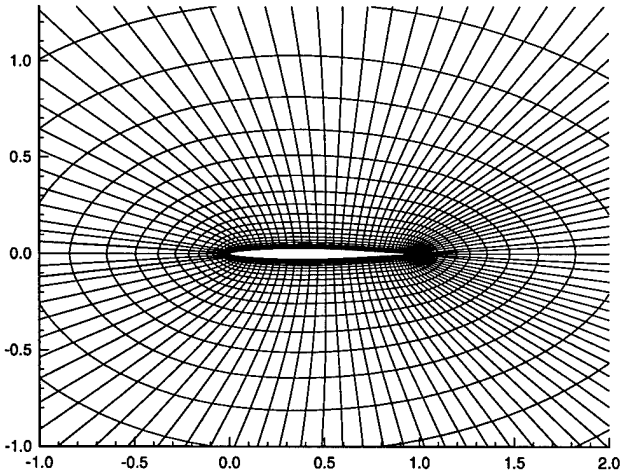


Figure 3.4 Node Distribution Near NACA 64A006 Airfoil (Representative Grid)

3.5 Boundary Conditions

A mapping is performed from the $x - y$ physical domain to the $\xi - \eta$ computational domain, as depicted in Figure 3.5. Boundaries A and C of Figure 3.5 are referred to as *cuts* and actually are not boundaries in the physical domain. Boundary B is broken up into three segments; B1 and B3 are inflow boundaries and B2 is an outflow boundary. The ranges of B1, B2, and B3 are governed by the sign of the contravariant velocity component normal to the grid:

$$\mathcal{V} = \eta_x u + \eta_y v. \tag{3.15}$$

Grid points with $\mathcal{V} > 0$ (which may change over the course of the calculation) are in segment B2 and the other points on boundary B are in either B1 or B3 depending on their location. Boundary D defines the airfoil surface. The conditions enforced along these boundaries is the subject of this section.

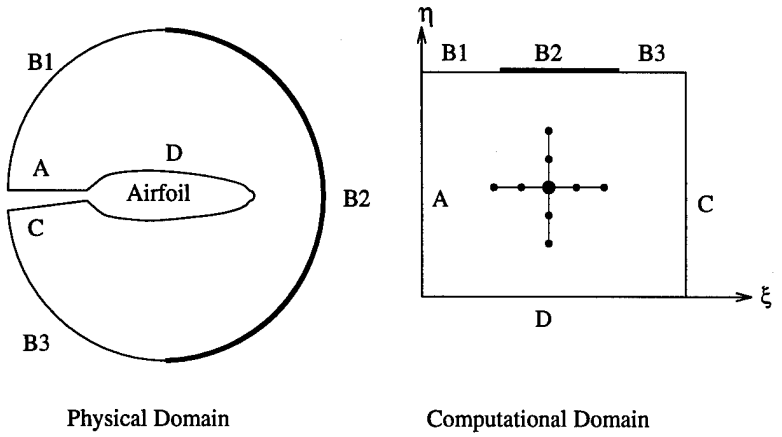


Figure 3.5 Relationship Between the Physical and Computational Domains

3.5.1 Farfield Inflow Boundary Conditions. The inflow conditions related to surfaces B1 and B3 are specified to be freestream conditions. Therefore, the governing equation along B1 and B3 can be written as

$$\bar{U}^{n+1} = \bar{U}_\infty, \tag{3.16}$$

where

$$\bar{U}_\infty = \begin{bmatrix} 1 \\ 1 \\ 0 \\ \frac{1}{\gamma(\gamma-1)M_\infty^2} + \frac{1}{2} \end{bmatrix}. \tag{3.17}$$

The discrete form of (3.16) is written in a delta form to maintain consistency between the discrete boundary condition and the interior equations:

$$\bar{U}_{i,J}^{n+1} = \bar{U}_{i,J}^{n} - \Delta t \left[\frac{1}{\Delta t} (\bar{U}_{i,J}^{n} - \bar{U}_\infty) \right], \quad (1 \le i \le I \text{ and } \mathcal{V} \le 0). \tag{3.18}$$

3.5.2 Farfield Outflow Boundary Conditions. Three of the outflow conditions are evolutionary forms of the conditions $\rho_x = 0$, $(\rho u)_x = 0$, and $(\rho v)_x = 0$:

$$f_t + u f_x = 0, \tag{3.19}$$

where $f = [\rho,\ \rho u,\ \rho v]^T$. The fourth outflow condition is a specification of total energy to be the freestream value. The form of the discrete equations is

$$f_{i,J}^{n+1} = f_{i,J}^n - \Delta t \left[\frac{u_{i,J}^n}{2} \left\{ \xi_{x\,i,J} \Delta f_{i,J}^n + \eta_{x\,i,J} (3 f_{i,J}^n - 4 f_{i,J-1}^n + f_{i,J-2}^n) \right\} \right], \tag{3.20}$$

$$E_{t\,i,J}^{n+1} = E_{t\,i,J}^n - \Delta t \left[\frac{1}{\Delta t} (E_{t\,i,J}^n - E_{t\,\infty}) \right], \tag{3.21}$$

for $1 \le i \le I$ and $\mathcal{V} > 0$. The form of Δf is

$$\Delta f_{i,J}^n = \begin{cases} 3 f_{i,J}^n - 4 f_{i-1,J}^n + f_{i-2,J}^n & \text{for } \xi_{x\,i,J} \ge 0 \\ -3 f_{i,J}^n + 4 f_{i+1,J}^n - f_{i+2,J}^n & \text{for } \xi_{x\,i,J} < 0 \end{cases}. \tag{3.22}$$

3.5.3 Airfoil Surface Boundary Conditions. The boundary condition at the airfoil surface appropriate for the Euler equations is flow tangency. Figure 3.6 shows the relationship between the normal and tangential velocities at the airfoil surface, u_s' and v_s', and the primitive variable velocities, u_e and v_e in the airfoil fixed coordinate system. Flow tangency (or impermeability) specifies the condition on the surface $v_s' = 0$. The remaining three conditions are ad-hoc conditions used commonly in the literature:

$$\frac{\partial u_s'}{\partial n} = 0, \quad \frac{\partial p}{\partial n} = 0, \quad \frac{\partial T}{\partial n} = 0. \tag{3.23}$$

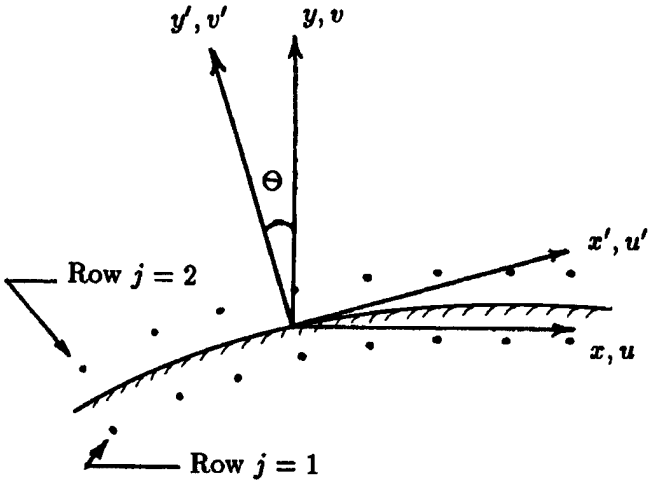


Figure 3.6 Grid Point-Surface Geometry

Manipulations of (3.23) produce conditions on the conserved variables

$$\frac{\partial \rho}{\partial n} = 0, \quad \frac{\partial E_t}{\partial n} = 0. \tag{3.24}$$

In summary, the airfoil surface boundary conditions are

$$v'_s = 0, \quad \frac{\partial u'_s}{\partial n} = 0, \quad \frac{\partial \rho}{\partial n} = 0, \quad \frac{\partial E_t}{\partial n} = 0. \tag{3.25}$$

The airfoil surface boundary conditions are implemented by assuming the surface lies between the first and second row of points (as depicted in Figure 3.6). Applying central-difference approximations to the Neumann conditions in (3.25), the following second-order-accurate discrete equations are obtained:

$$\rho_{i,1} = \rho_{i,2}, \quad E_{ti,1} = E_{ti,2}, \quad u'_{i,1} = u'_{i,2}. \tag{3.26}$$

Expressing the surface velocity in terms of the points adjacent to the surface (Figure 3.6), the following first-order-accurate expression is obtained:

$$v'_{si} = \frac{1}{2}(v'_{i,1} + v'_{i,2}). \tag{3.27}$$

Applying the impermeability condition of (3.25), (3.27) becomes

$$v'_{i,1} = -v'_{i,2}. \tag{3.28}$$

The conditions on velocity in (3.26)-(3.28) can be rewritten as

$$\begin{pmatrix} u'_{i,1} \\ v'_{i,1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} u'_{i,2} \\ v'_{i,2} \end{pmatrix}. \tag{3.29}$$

The variables u_e and v_e are evaluated at the first row of nodes ($j = 1$) with a coordinate rotation as depicted in Figure 3.6 [93]. The relationship between (u_e, v_e) and (u', v') is

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} \cos\bar{\Theta} & \sin\bar{\Theta} \\ -\sin\bar{\Theta} & \cos\bar{\Theta} \end{pmatrix} \begin{pmatrix} u_e \\ v_e \end{pmatrix}, \tag{3.30}$$

where $\bar{\Theta}$ is the local slope angle of the body. Combining (3.29) and (3.30), (3.29) becomes

$$\begin{pmatrix} \cos\bar{\Theta} & \sin\bar{\Theta} \\ -\sin\bar{\Theta} & \cos\bar{\Theta} \end{pmatrix} \begin{pmatrix} u_{ei,1} \\ v_{ei,1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \cos\bar{\Theta} & \sin\bar{\Theta} \\ -\sin\bar{\Theta} & \cos\bar{\Theta} \end{pmatrix} \begin{pmatrix} u_{ei,2} \\ v_{ei,2} \end{pmatrix}, \tag{3.31}$$

which can be rewritten as

$$\begin{pmatrix} u_{ei,1} \\ v_{ei,1} \end{pmatrix} = \begin{pmatrix} t_1(\bar{\Theta}) & t_2(\bar{\Theta}) \\ t_2(\bar{\Theta}) & -t_1(\bar{\Theta}) \end{pmatrix} \begin{pmatrix} u_{ei,2} \\ v_{ei,2} \end{pmatrix}, \tag{3.32}$$

using the definitions

$$t_1(\beta) \equiv \cos^2\beta - \sin^2\beta, \quad t_2(\beta) \equiv 2\cos\beta\sin\beta, \tag{3.33}$$

for some arbitrary angle β.

The velocities on the surface are now in the airfoil fixed reference system. To obtain the inertial velocity components, denoted by a subscript in, the following relationship is used:

$$\begin{pmatrix} u_e \\ v_e \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} u_{in} \\ v_{in} \end{pmatrix}. \tag{3.34}$$

Substituting (3.34) into (3.31), multiplying by the density, and operating on the result with the inverse of the direction cosine matrix, the variables $(\rho u)_{i,1}$ and $(\rho v)_{i,1}$ are found to be

$$\begin{pmatrix} (\rho u)_{i,1} \\ (\rho v)_{i,1} \end{pmatrix} = \begin{pmatrix} t_1(\alpha)t_1(\bar{\Theta}) + t_2(\alpha)t_2(\bar{\Theta}) & t_1(\alpha)t_2(\bar{\Theta}) - t_2(\alpha)t_1(\bar{\Theta}) \\ t_1(\alpha)t_2(\bar{\Theta}) - t_2(\alpha)t_1(\bar{\Theta}) & -t_1(\alpha)t_1(\bar{\Theta}) - t_2(\alpha)t_2(\bar{\Theta}) \end{pmatrix} \begin{pmatrix} (\rho u)_{i,2} \\ (\rho v)_{i,2} \end{pmatrix}, \tag{3.35}$$

where the boundary condition, $\rho_{i,1} = \rho_{i,2}$ is used to maintain only $(i,2)$ data on the right hand side. The equations for $\rho_{i,1}$ and $E_{ti,1}$ in (3.26) and equation (3.35) yield a complete set of surface boundary conditions valid for $1 \leq i \leq I$.

3.5.4 Cut Boundary Conditions. The cut boundaries A and C are interior lines of nodes in the physical domain (see Figure 3.5). The interior equations, (3.11)-(3.13), are applied along these boundaries to allow consistency with the interior equations. Unfortunately, the stencil of conserved variables extends outside of the domain for these boundaries and for interior equations with $i = 2$ and $i = I - 1$. The stencil variables with indices which have no physical meaning are mapped to values in the domain in the following way:

$$\bar{U}_{-1,j} = \bar{U}_{I-1,j}, \quad \bar{U}_{0,j} = \bar{U}_{I,j}, \quad \bar{U}_{I+1,j} = \bar{U}_{1,j}, \quad \bar{U}_{I+2,j} = \bar{U}_{2,j}, \tag{3.36}$$

for all $1 \leq j \leq J$.

3.6 Calculation of Force and Moment Coefficients

The force and moment coefficients are computed with numerical integration of the surface pressures. The pressure at a node on the surface is computed with a simple average of the neighboring pressures

$$p_{si} = \frac{1}{2}(p_{i,1} + p_{i,2}). \tag{3.37}$$

The pressure along the surface between two nodes is assumed to be constant and given by

$$p_{mid_i} = \frac{1}{2}(p_{si} + p_{si+1}). \tag{3.38}$$

The components of the distance between two surface nodes is

$$dx_i = \frac{1}{2}(x_{i+1,1} + x_{i+1,2}) - \frac{1}{2}(x_{i,1} + x_{i,2}), \quad dy_i = \frac{1}{2}(y_{i+1,1} + y_{i+1,2}) - \frac{1}{2}(y_{i,1} + y_{i,2}). \tag{3.39}$$

The lift coefficient can then be expressed as [138]

$$C_l = -2\cos\alpha \sum_{i=1}^{I} p_{mid_i} dx_i - 2\sin\alpha \sum_{i=1}^{I} p_{mid_i} dy_i. \tag{3.40}$$

The drag coefficient can be written as [138]

$$C_d = 2\cos\alpha \sum_{i=1}^{I} p_{mid_i} dy_i - 2\sin\alpha \sum_{i=1}^{I} p_{mid_i} dx_i. \tag{3.41}$$

The moment coefficient about the cg is expressed as [138]

$$C_m = 2\sum_{i=1}^{I} p_{mid_i} \left(dx_i x_{cg} + dy_i y_{cg}\right). \tag{3.42}$$

The subscript $i = I + 1$ implies $i = 1$ in (3.40)-(3.42).

3.7 Equilibrium System

The discrete form of the governing equations discussed in Chapter II is defined

$$U_t = F(U). \tag{3.43}$$

Equations (3.11)-(3.13) are placed in this form by defining

$$F_k \equiv \frac{\bar{U}_{i,j}^{n+1} - \bar{U}_{i,j}^{n}}{\Delta t}, \tag{3.44}$$

for $1 \leq i \leq I$ and $2 \leq j \leq J-1$. This implies that F_k for the interior and cut equations becomes

$$F_k = (L_\xi L_\eta \hat{U}_{i,j}^{n} - \hat{U}_{i,j}^{n})\frac{\mathcal{J}_{i,j}}{\Delta t} = -\mathcal{J}_{i,j}\left[(\tilde{\hat{G}}_{i,j+\frac{1}{2}}^{n} - \tilde{\hat{G}}_{i,j-\frac{1}{2}}^{n}) + \left(\tilde{\hat{F}}_{i+\frac{1}{2},j}^{n+1/2} - \tilde{\hat{F}}_{i-\frac{1}{2},j}^{n+1/2}\right)\right]. \tag{3.45}$$

Applying the assumption of equilibrium, $F_k = 0$. F_k is an $\mathcal{R}^4$ vector and the index k of F_k determines the collocation strategy and is defined from

$$k \equiv (i-1)J + j, \quad \text{for } 1 \leq i \leq I, \;\; 2 \leq j \leq J, \tag{3.46}$$

for the column-row strategy, (i.e., indexing by columns) employed for this research. The farfield boundary conditions are placed in the same form and are defined with

$$F_k = \frac{\bar{U}_{i,J}^{n+1} - \bar{U}_{i,J}^{n}}{\Delta t} = -\left[\frac{1}{\Delta t}(\bar{U}_{i,J}^{n} - \bar{U}_\infty)\right], \tag{3.47}$$

for $1 \leq i \leq I$, and $\mathcal{V} \leq 0$, (boundaries B1 and B3). The outflow outer boundary equilibrium equations are defined with

$$F_k = \frac{\bar{U}_{i,J}^{n+1} - \bar{U}_{i,J}^{n}}{\Delta t} = \begin{bmatrix} -\frac{u_{i,J}^n}{2}\left\{\xi_{xi,J}\Delta f_{i,J}^n + \eta_{xi,J}(3f_{i,J}^n - 4f_{i,J-1}^n + f_{i,J-2}^n)\right\} \\ -\frac{1}{\Delta t}(E_{ti,J}^n - E_{t\infty}) \end{bmatrix}, \tag{3.48}$$

for $1 \leq i \leq I$, and $\mathcal{V} > 0$ (boundary B2). The surface boundary conditions, (3.26) and (3.34), are used to compute the fluxes for the first row of nodes inside the domain and are not included in the fully implicit system. The set of $\mathcal{R}^4$ unknown vectors, $\bar{U}_{i,j}$, and equilibrium equations, F_k, are placed in the $\mathcal{R}^N$ vectors U and F respectively, by incrementing k from 1 to the total number of equations, N:

$$N = 4I(J-1). \tag{3.49}$$

The equilibrium system is then described by

$$F(U;\tilde{\lambda}) = 0, \tag{3.50}$$

where $\tilde{\lambda}$ is some free parameter of the system, such as M_∞, and is solved with Newton's method (described in Chapter II).

The Jacobian matrix, F_U, is an integral part of the Newton's method solution process. The equations near the cut access conserved variables across the cut, whereas the conserved variables of the remaining equations are all within a narrow *bandwidth* of each other. For this reason, the equations near the cut are blocked off from the other equations, resulting in a Jacobian matrix of the form

$$F_U = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \tag{3.51}$$

where A is a banded system with a bandwidth, A_{bw}, given by

$$A_{bw} = 16(J-1)+25. \tag{3.52}$$

If a row-column collocation strategy is employed, J in (3.52) is replaced with I. It is apparent that since J is usually a fraction of I, the bandwidth is smaller for a column-row collocation strategy. The number of elements of F in the cut region is found to be

$$N_{cut} = 8(J-1). \tag{3.53}$$

The row $\times$ column dimensions of A, B, C, and D are then

$$A: \quad (N-N_{cut}) \times (N-N_{cut}), \tag{3.54}$$

$$B: \quad (N-N_{cut}) \times (N_{cut}), \tag{3.55}$$

$$C: \quad (N_{cut}) \times (N-N_{cut}), \tag{3.56}$$

$$D: \quad (N_{cut}) \times (N_{cut}). \tag{3.57}$$

Chapter II presents the linear system describing a single Newton iterate, (2.18). A method of solving (2.18) efficiently for the Jacobian matrix structure of (3.51) is described in Appendix C and is employed in the current research.

The original version of TVDntiAE, the current research computer code, employs a row-column strategy, since with row-column indexing the blocked linear system solver of Appendix C is not necessary. With row-column indexing, the size of I defines the bandwidth. A "C"-grid topology has on the order of twice the number of elements in I as a result of the region aft of the airfoil, while maintaining a similar J. Thus, the bandwidth of the Jacobian is therefore larger for a "C"-grid than that of A of an "O"-grid. This additional bandwidth is the original basis for choosing an

"O"-grid. In addition, a "C"-grid has clustering in the wake region of the airfoil as a result of the topology. Since the Euler equations are used for the current research, grid point clustering in the wake region is not necessary and therefore adds to the computational resource requirements.

IV. Static Airfoil Results

With a computer code based on the model described in Chapter III (called TVDntiAE), code validation is accomplished. The validation procedure is a comparison of solutions for a variety of grids and parameters of the algorithm. Also, a comparison of the solutions obtained with the current method and a solution from the open literature is made. Next, convergence properties of the explicit method and the equilibrium method are documented, as well as the consistency between the two methods. Finally, equilibrium solutions are presented for NACA 0012 and NACA 64A006 airfoils at various Mach number and angle-of-attack conditions. These solutions are included to document the code's ability to compute solutions for ranges of M_∞ and α.

4.1 Algorithm Validation (Equilibrium Solutions)

This section presents the algorithm validation comprised of a sensitivity analysis and comparison with an AGARD [140] study computed solution from the open literature for a static NACA 0012 airfoil. All flowfield solutions are for $M_\infty = 0.8$ and $\alpha = 1.25^o$ to provide consistency with the AGARD solution.

4.1.1 Grid-Sensitivity Study. The effects of grid refinement are presented for variations in domain size, R_{max}, wall spacing, $\Delta wall$, the number of nodes in the normal direction, J, and the number of nodes around the airfoil, I. Also, the TVD parameter δ_1 and the Courant number, ν_{cfl}, defined in Appendix A, are varied and the effects on the flowfield solution documented. Table 4.1 describes all of the grids and parameter values used in the sensitivity study, each of which is assigned a case number. Solutions for cases 1-17, were obtained with the Newton-TVD algorithm converged to an L_2 norm of the residual, $||F||_2$, less than ϵ_{conv} (10^{-12}). The coefficient of lift, C_l, is used as the sensitivity metric in the validation study. Also, the effect of varying grid and TVD parameters on the surface coefficient of pressure, C_p, is documented.

Case#	*Grid#*	*I*	*J*	R_{max}	Δ_{wall}	δ_1	ν_{cfl}	C_l	Type
1	G12-1	80	160	150	0.001	0	0.5	0.3452	TI
1a	G12-1	80	120	100	0.001	0	0.5	0.3432	EQ
2	G12-2	80	80	50	0.001	0	0.5	0.3286	EQ
3	G12-3	80	64	25	0.001	0	0.5	0.3211	EQ
4	G12-4	80	60	20	0.001	0	0.5	0.3173	EQ
5	G12-5	80	56	15	0.001	0	0.5	0.3084	EQ
6	G12-6	80	42	10	0.001	0	0.5	0.2997	EQ
7	G12-7	80	42	10	0.003	0	0.5	0.2935	EQ
8	G12-8	80	42	10	0.005	0	0.5	0.2877	EQ
9	G12-9	80	32	10	0.005	0	0.5	0.2871	EQ
10	G12-10	80	16	10	0.005	0	0.5	0.2700	EQ
11	G12-11	160	32	10	0.005	0	0.5	0.2954	EQ
12	G12-12	120	32	10	0.005	0	0.5	0.2918	EQ
13	G12-13	80	32	10	0.005	0.01	0.5	0.2871	EQ
14	G12-14	80	32	10	0.005	0.1	0.5	0.2871	EQ
15	G12-15	80	32	10	0.005	0.5	0.5	0.2871	EQ
16	G12-16	80	32	10	0.005	0	0.4	0.2871	EQ
17	G12-17	80	32	10	0.005	0	0.6	0.2871	EQ
18	G12-18	320	64	25	0.001	0	0.5	0.3432	TI

Table 4.1 Case Definitions for Sensitivity Analysis

First, the domain size, R_{max}, is varied to determine its effects on the flowfield solution. The domain size is varied from a radius at the farfield boundary of 150 chord lengths to 10 chord lengths (cases 1-6), by excluding all points outside of the desired farfield radius. This method of producing grids has the advantage of holding all other grid parameters fixed. Figure 4.1a displays the surface coefficient of pressure versus the coordinate along the chord line for various domain sizes. The difference in pressure coefficient is primarily confined to three points, typical of TVD algorithms. The largest difference is found in the upper surface shock at the middle point. Shock capturing methods do not resolve the shock completely and, therefore, errors at the middle point are expected, especially when first-order dissipation is applied at the shock. The first and last points are practically the same for all domain sizes with the middle point having a large variation. As R_{max} increases, C_p of the middle point increases, trending toward matching the upstream state of the shock. This effectively moves the upper shock position aft by one grid point. The strength of the lower surface shock decreases and the position moves forward as R_{max} increases.

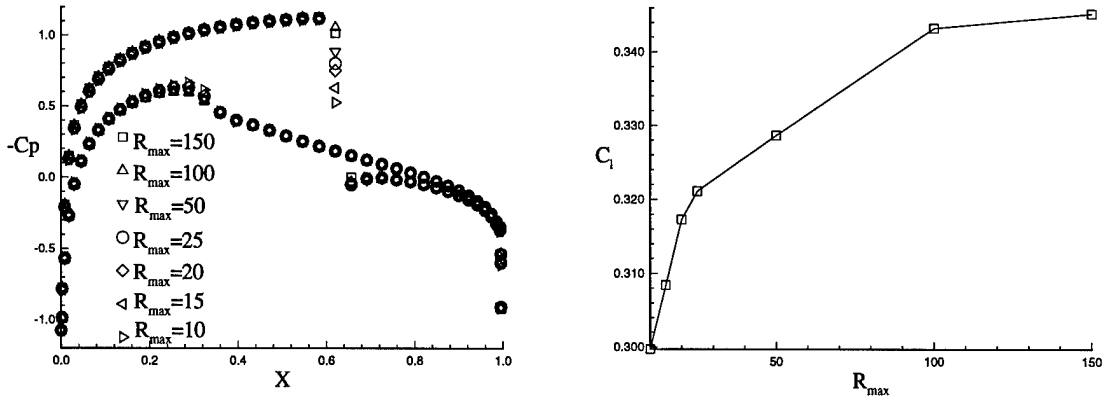


Figure 4.1 Sensitivity to Outer Domain Size

The coefficient of lift varies by 13.2% from the largest to smallest domain size. Figure 4.1 depicts the variation of C_l with domain size. A definite slope change in $C_l(R_{max})$ is observed between $R_{max} = 20$ and $R_{max} = 25$ chord lengths.

The effects of wall spacing on the flowfield solution are next examined. The surface is assumed to be between the first two normal nodes. Since the surface boundary conditions consist of Neumann conditions in discrete form, which are a function of $\Delta wall$, a strong sensitivity to $\Delta wall$ is expected. The wall spacing is varied from 0.001 to 0.003 and then to 0.005 in cases 6-8 with $R_{max} = 10$. Figure 4.2a displays the surface coefficient of pressure for the three wall spacings. The various solutions are in good agreement for the majority of the points. The largest difference is seen at the leading-edge and trailing-edge points. A moderate difference in pressure coefficient can be seen for two points in the embryonic shock on the lower surface. The coefficient of lift varies by 4.0% from the coarsest to finest wall spacings. Figure 4.2b shows that C_l monotonically increases as $\Delta wall$ goes to zero.

In cases 8-10, the number of nodes in the normal direction is varied from 16 to 32 and then to 42 to determine the effects of radial refinement, holding $R_{max} = 10$ and $\Delta wall = 0.005$. Figure 4.3a shows the pressure coefficient comparison. Solutions for $J = 42$ and $J = 32$ show

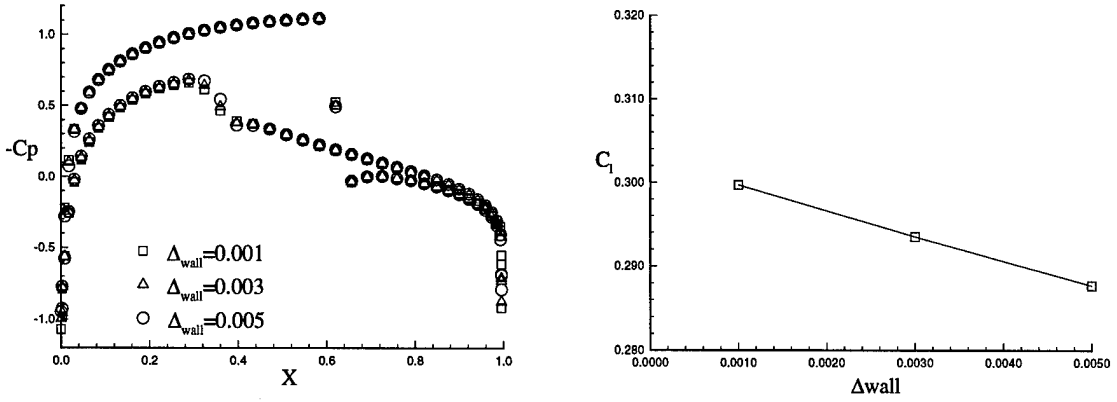


Figure 4.2 Sensitivity to Wall Spacing

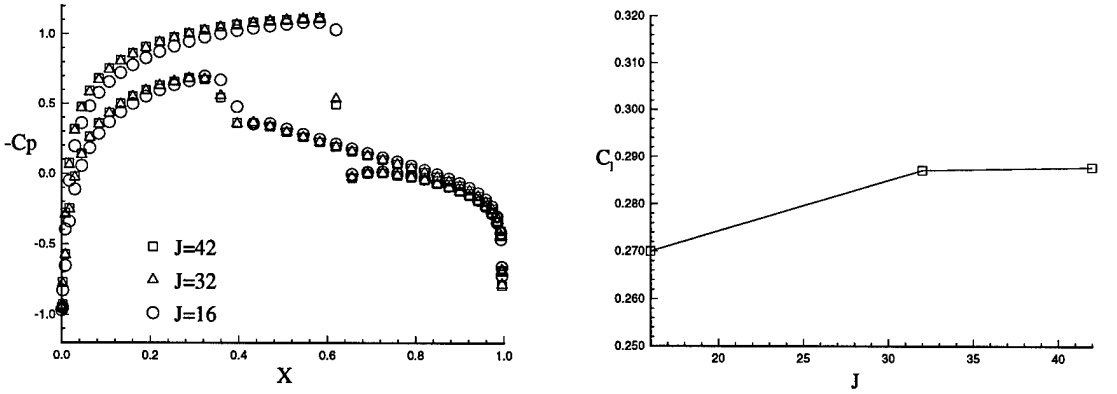


Figure 4.3 Sensitivity to Node Quantity in Normal Direction

no practical differences with a lift coefficient variation of only 0.2%. The solution for $J = 16$ is different from the other two solutions in the expansion regions above and below the airfoil. Also, the shock positions on the upper and lower surfaces vary by approximately one grid point or 0.04 chord lengths. The overall variation in lift coefficient for the three grids is 6.1%. As J increases, $C_l(J)$ increases (Figure 4.3b).

The node distribution around the airfoil is varied to determine the effect on the flowfield solution. The number of nodes is varied from 80 to 120 and then to 160 with cases 9, 11 and 12. The majority of the difference in pressure coefficient is seen at the upper and lower shock and at

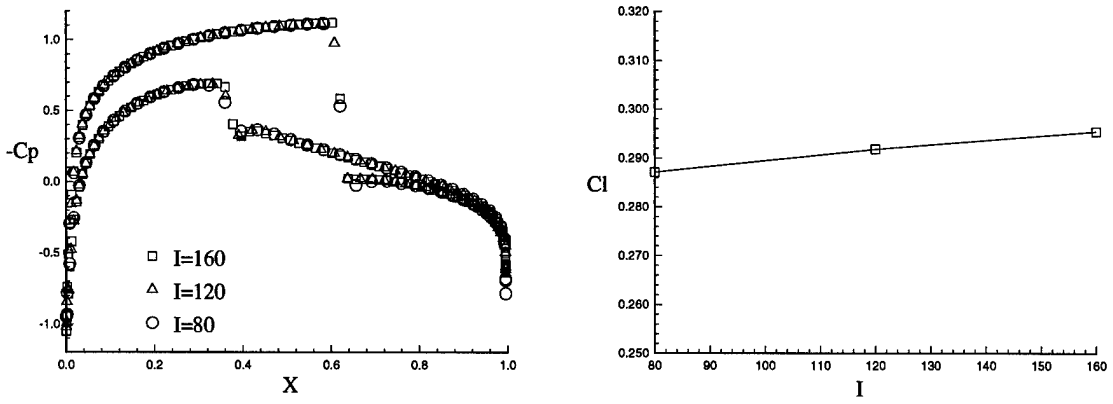


Figure 4.4 Sensitivity to Node Quantity Around the Airfoil

the trailing edge point. All three solutions predict the same upper shock location and strength with a variation in the shock middle point. The lower shock strength and position varies slightly with a trend toward a stronger shock with refinement. The variation in the lift coefficient for the three solutions is only 2.8%, with a monotonically increasing behavior as I is increased (Figure 4.4b).

In addition to the grid-sensitivity study, the sensitivities of the flowfield solution to the TVD entropy correction parameter, δ_1 and the maximum (or critical) Courant number, ν_{cfl}, are presented. The TVD parameter δ_1 is varied from 0 to 1/2 in cases 9 and 13-15, resulting in a minimal variation in C_l of 0.2%. Since δ_1 has minimal effect on the solution, a value of zero is used for all other validation cases. This value of δ_1 is consistent with the Roe scheme. ν_{cfl} is varied from 0.4 to 0.5 and then to 0.6 in cases 9, 16 and 17, resulting in a 1.1% variation in C_l. Since ν_{cfl} had minimal effect on the solution, and $\nu_{cfl} = 1/2$ provided the best convergence of the TVD time-integration algorithm, this value is used for all other validation cases.

Results of all sensitivity studies are summarized in Table 4.2. Variation of parameter values in the direction of improved refinement and increased domain size always increased C_l. Overall, comparison between the best and worst cases (1 and 10) shows a 22% variation in C_l. The parameter of greatest sensitivity is seen to be the domain size, R_{max}. When R_{max} is decreased by 93% from

Parameter	Min-Max Values	C_l Variation
R_{max}	10-100	13.2%
Δ_{wall}	0.001-0.005	4.0%
J	16-42	6.1%
I	80-16	2.8%
δ_1	0-0.5	0.2%
ν_{cfl}	0.4-0.6	1.1%
All	Worst-Best	22%

Table 4.2 Summary of Sensitivity Evaluations

its maximum value, C_l is decreased by 13.2%. Furthermore, the results indicate that C_l asymptotes only for $R_{max} > 150$. R_{max} is judged to be the most limiting parameter, in terms of accuracy.

4.1.2 Comparison with AGARD Solution. Further validation of the method is accomplished through comparison with a solution from the open literature. An AGARD solution [140], which is deemed the best of a set of solutions by the authors of the study, is used to provide the comparison. To compare directly (i.e., with similar grids) the current method with the AGARD solution, grid G12-18 with $\delta_1 = 0$ and $\nu_{cfl} = 1/2$ is employed. Due to resource limitations associated with the Newton-TVD method, the TVD time-integration algorithm is used, converged to an L_2 norm of 10^{-5} The larger cutoff criterion is due to poor convergence properties of the time-integration method. Section 4.2 presents a discussion of the algorithm convergence properties as well as a discussion of the acceptable ϵ_{conv}. Figure 4.5 presents the coefficient of pressure distribution on the surface for the TVD time-integration solution (case 18), the solutions of cases 7 and 10, and the AGARD data set. The upper surface shock is captured in three grid points for cases 7, 10, and 18, whereas the AGARD solution requires five points. The location of the AGARD shock is slightly downstream of the solutions from the current method. As described in Section 4.1.1, the trends in TVDntiAE solutions for an increasing domain size are the upper surface shock moves aft, and the lower surface shock weakens and moves forward. It is apparent from the tabulated AGARD data [140] that the conditions (3.17) are not enforced at the farfield boundary. The boundary conditions of the AGARD solution can cause an effectively larger domain size and

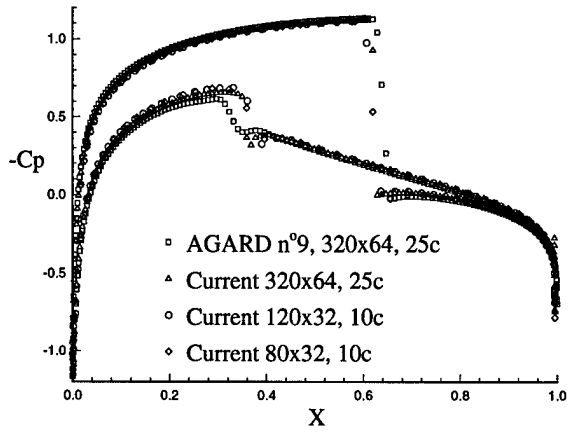


Figure 4.5 Surface Cp: $M_\infty = 0.8$ and $\alpha = 1.25^o$

therefore have a more aft shock on the upper surface and a weaker forward shock on the lower surface. There is also an offset in the expansion regions of the upper and lower surfaces, resulting in a higher overall lift for the AGARD solution than for the current solutions. The most refined TVDntiAE solution predicts $C_l = 0.3452$, whereas for the AGARD solution, $C_l = 0.3632$ (a 5.2% difference). Figure 4.1b shows a trend of increasing domain size increasing to the AGARD solution. Since it is not practical computationally to continue the domain size study, no further attempt is made to determine the asymptotic value of C_l using the TVD time-integration method.

4.2 Convergence Properties of Newton-TVD and TVD-Time-Integration Algorithms

In this section, steady-state solutions are computed for a NACA 64A006 airfoil using grid G646-2 with both algorithms to demonstrate their convergence properties. Also, the convergence properties are investigated for a variation in the numerical Jacobian parameter, ϵ_{jac}. The flowfield conditions are $M_\infty = 0.85$ and $\alpha = 0^o$.

4.2.1 Newton-TVD and TVD-Time-Integration Comparison. A comparison of the Newton-TVD and TVD-time-integration methods is provided to describe the consistency in the solutions

Case#	*Grid#*	M_∞	α	ϵ_{jac}	Type
19	G646-2	0.85	0°	N/A	TI
20	G646-2	0.85	0°	10^{-5}	EQ
21	G646-2	0.85	0°	10^{-3}	EQ
22	G646-2	0.85	0°	10^{-4}	EQ
23	G646-2	0.85	0°	10^{-6}	EQ
24	G646-2	0.85	0°	10^{-7}	EQ
25	G646-2	0.85	0°	10^{-8}	EQ

Table 4.3 Convergence Properties Run Summary

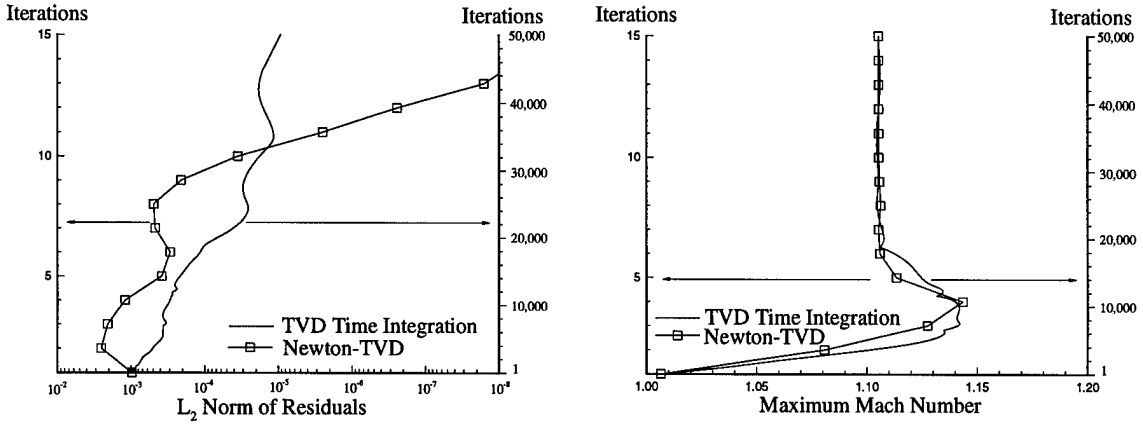


Figure 4.6 Convergence Histories of Newton-TVD and TVD-Time-Integration Algorithms

and their respective convergence properties. The flowfield is initialized with freestream values and then the time-integration algorithm is used to compute a solution to a residual norm of 10^{-3}. This solution is then used to initialize both the Newton-TVD and TVD-time-integration algorithms and solutions are computed. The numerical Jacobian parameter for the Newton-TVD algorithm is $\epsilon_{jac} = 10^{-5}$. Figure 4.6 depicts the convergence histories for the two algorithms. The TVD-time-integration method converges with a linear rate typical of explicit TVD schemes in 50,000 iterations to a residual norm of 9×10^{-6}. In approximately 20,000 iterations the peak Mach number has reached its asymptotic value of 1.105, corresponding to a residual norm of 6×10^{-5}. The Newton-TVD algorithm computes a solution in 14 iterations to a residual norm less than 10^{-8}. The asymptotic value of peak Mach number, 1.105, is reached in 10 iterations, corresponding to a residual norm of 3×10^{-4}. The convergence rate is not formally quadratic, but it is linear with

the number of iterations on the order of a quadratic method. The author attributes this degraded convergence behavior to two constraints. First, the computed strong shock violates one of the Newton's method criteria of continuous differentiability [151]. Second, the Jacobian elements are not exact; they are numerical approximations of the analytical Jacobian elements. The two solutions are considered consistent with each other when considering the peak Mach number as the comparison metric.

4.2.2 Variation of Numerical Jacobian Parameter. The effects of the non-exact numerical Jacobian elements are documented in this subsection. The details of computing numerical Jacobian elements are provided in Appendix B. The numerical Jacobian is a second-order-accurate finite-difference approximation. The parameter, ϵ_{jac}, defines the order-of-error of the approximation and is also the divisor of the finite-difference approximation. To document the effects of ϵ_{jac}, solutions are computed for $10^{-3} \leq \epsilon_{jac} \leq 10^{-8}$ and depicted in Figure 4.7. The convergence history for $\epsilon_{jac} = 10^{-5}$ has the best slope of all solutions computed and reaches machine precision at approximately 20 iterations. The convergence history for $\epsilon_{jac} = 10^{-3}$ asymptotes to a residual norm greater than 10^{-6}, which is much larger than machine precision, displaying the effect of truncation error on the Jacobian elements. When $\epsilon_{jac} = 10^{-4}$, the solution does not converge in twenty iterations, but appears to be converging towards the solution for $\epsilon_{jac} = 10^{-5}$. The convergence history for $\epsilon_{jac} = 10^{-6}$ is degraded from the $\epsilon_{jac} = 10^{-5}$ history and reaches a residual norm of 10^{-12} in twenty iterations. The $\epsilon_{jac} = 10^{-7}$ and $\epsilon_{jac} = 10^{-8}$ histories again asymptote to a residual norm of 10^{-6} due to round-off error. The optimum ϵ_{jac} is found to be 10^{-5} and is used for all calculations in this research unless otherwise specified.

4.3 Variation of Mach Number and Angle-of-Attack

This section presents equilibrium solutions for the two airfoils, NACA 0012 and NACA 64A006, at various Mach numbers and angles-of-attack. The purpose of the solutions computed in

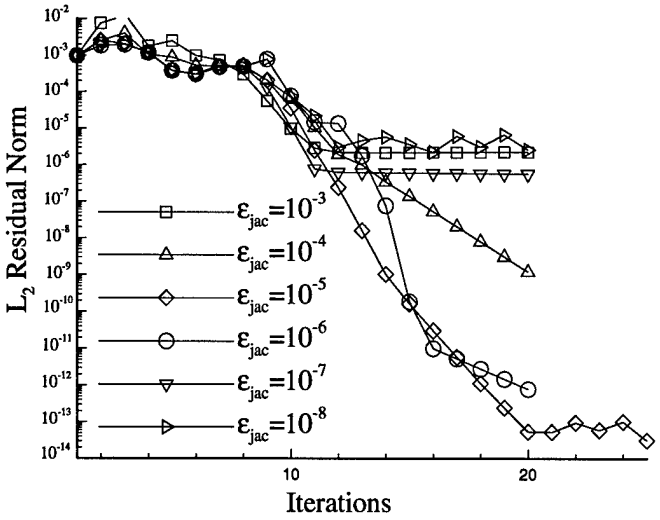


Figure 4.7 Convergence Histories for Various Numerical Jacobian Accuracies (Newton-TVD)

this section is to provide comparison and demonstration data for the chosen numerical algorithm. The NACA 0012 solutions are computed with grid G12-13 and the NACA 64A006 solutions are computed with grid G646-1. For all solutions, $\delta_1 = 0$ and $\nu_{cfl} = 1/2$. The effects of varying the freestream Mach number and angle-of-attack on the flowfield are assessed using contour plots of the local Mach number. Also, C_l, C_d, and $C_{m_{1/4c}}$ versus angle-of-attack are presented for the NACA 64A006 airfoil.

4.3.1 NACA 0012. Solutions for various Mach numbers and angles-of-attack are presented for the NACA 0012 airfoil. The freestream Mach number is varied from 0.63 to 1.2; Mach contours are displayed in Figure 4.8. The angle-of-attack is held fixed at 1.25^o.

Figure 4.8a depicts the flowfield solution for $M_\infty = 0.63$. Supersonic flow is not evident for this freestream Mach number. The freestream Mach number is increased to 0.80 (Figure 4.8b). A strong shock is found on the upper surface slightly aft of the mid-chord, and an embryonic shock is observed on the lower surface near the quarter-chord. The freestream Mach number is increased to 0.95 and the the upper and lower shocks strengthen and move to the trailing edge (Figure 4.8c). The

Case#	*Grid#*	M_∞	α	Type
26	G12-13	0.63	1.25^o	EQ
27	G12-13	0.80	1.25^o	EQ
28	G12-13	0.95	1.25^o	EQ
29	G12-13	1.20	1.25^o	EQ
30	G12-13	0.80	0^o	EQ
31	G12-13	0.80	5.0^o	EQ
32	G12-13	0.80	10.0^o	EQ
33	G646-1	0.85	0^o	EQ
34	G646-1	0.85	1.25^o	EQ
35	G646-1	0.85	5.0^o	EQ
36	G646-1	0.85	10.0^o	EQ
37	G646-1	0.85	$0 - 5.75^o$	EQ

Table 4.4 Run Summary: Variation of M_∞ and α

supersonic region extends well into the domain. When the freestream Mach number is increased to 1.2, a bow shock forms in front of the airfoil and a pocket of subsonic flow develops between the bow shock and the leading edge of the airfoil (Figure 4.8d).

Next, the airfoil angle-of-attack is varied from 0 to 10 degrees, while fixing the freestream Mach number at 0.80. Mach contours are displayed in Figure 4.9. Figure 4.9a depicts the flowfield solution for $\alpha = 0^o$. Symmetric shocks are in evidence above and below the surface at approximately mid-chord. The supersonic region extends approximately a half-chord length into the domain. The angle-of-attack is increased to 1.25^o (Figure 4.9b), which is the same case displayed in Figure 4.8b. For an angle-of-attack of 5^o, the lower surface shock is not in evidence and the upper shock is farther aft and strengthened (Figure 4.9c). The supersonic region extends more than a chord length into the domain. The angle-of-attack is increased to 10^o and the shock position is not changed significantly but the shock strength increases and the supersonic region increases in size.

4.3.2 NACA 64A006. Solutions for various angles-of-attack are presented for the NACA 64A006. Mach contours for selected angles-of-attack (0^o, 1.25^o, 5^o, and 10^o) are computed and are displayed in Figure 4.10. The freestream Mach number is fixed at 0.85. Also, C_l, C_d and $C_{m_{1/4c}}$ versus angle-of-attack for a range of α is displayed for $M_\infty = 0.85$.

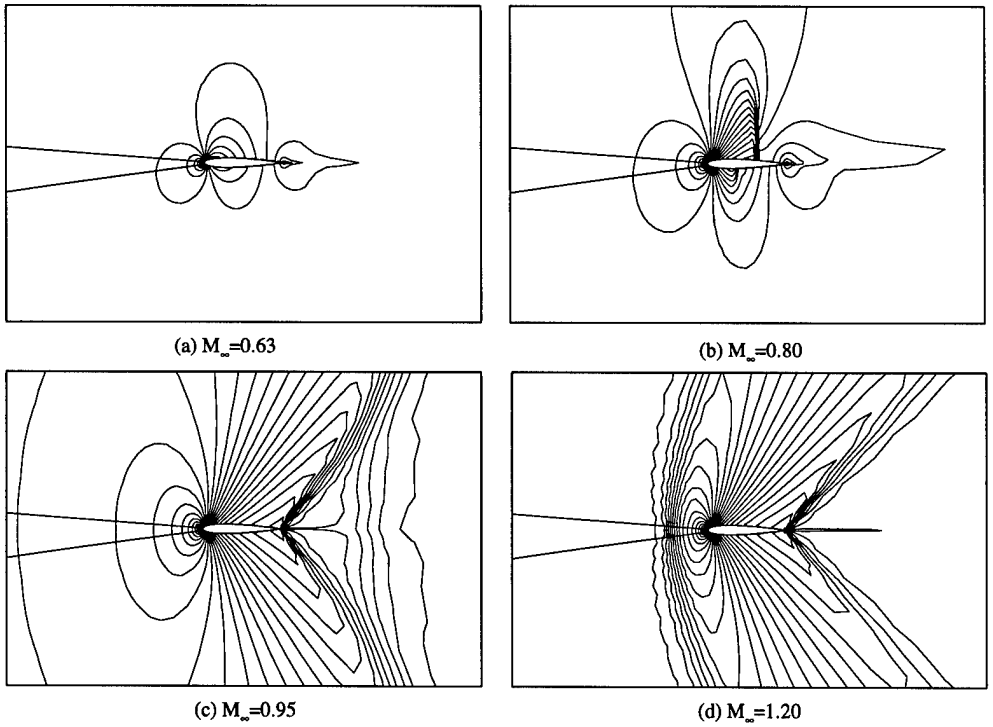


Figure 4.8 Effects of Freestream Mach Number on Mach Contours: NACA 0012, $\alpha = 1.25^o$

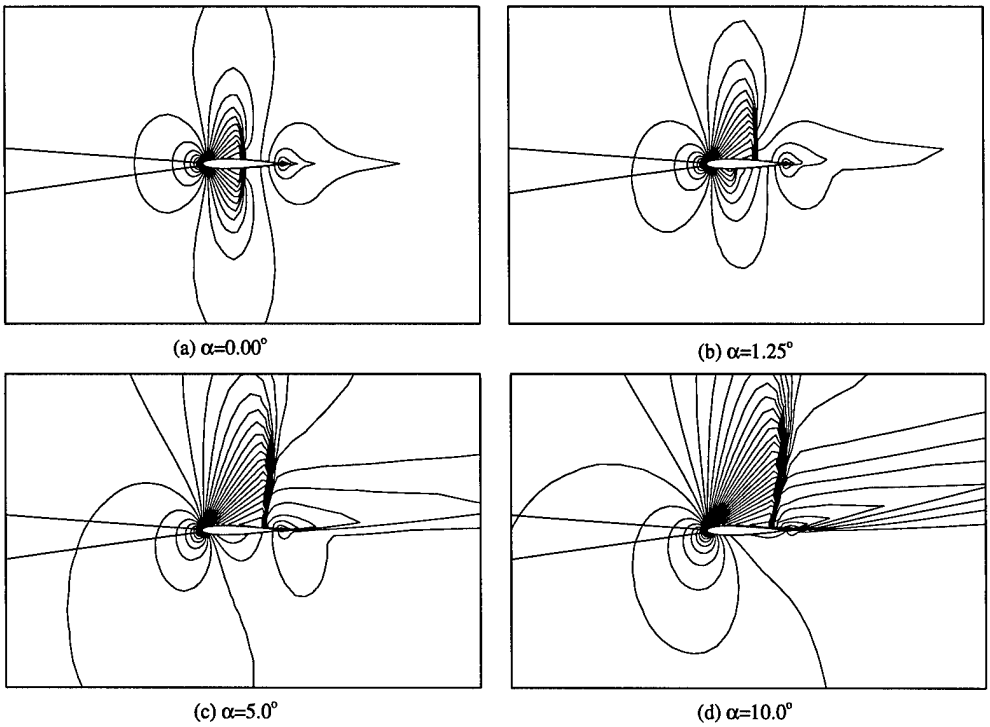


Figure 4.9 Effects of Angle-of-Attack on Mach Contours: NACA 0012, $M_\infty = 0.80$

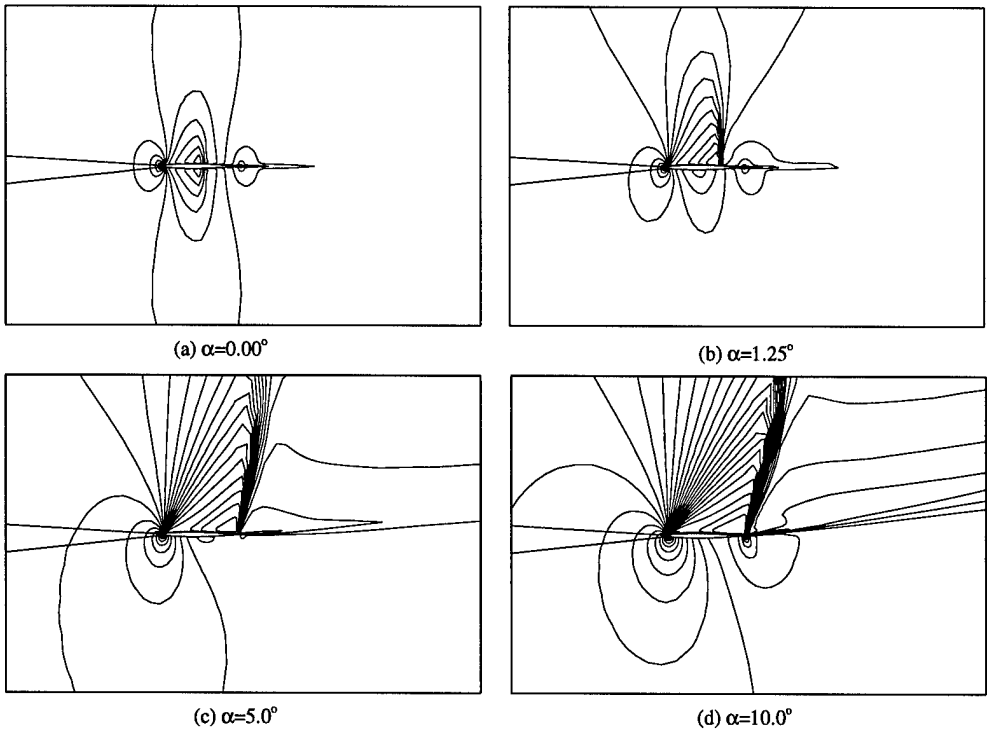


Figure 4.10 Effects of Angle-of-Attack on Mach Contours: NACA 64A006, $M_\infty = 0.85$

The case for $\alpha = 0^o$ has weaker shocks on the upper and lower surface than the NACA 0012 airfoil even with a larger freestream Mach number, due to the airfoil thickness change (Figure 4.10a). The angle-of-attack is increased to 1.25^o; the shock wave strengthens and moves aft on the upper surface and the lower surface shock is not evident. For $\alpha = 5^o$, the upper surface shock moves to the trailing edge and strengthens, resulting in a larger supersonic region above the surface (Figure 4.10c). The angle-of-attack is increased to 10^o and the upper surface shock wave strengthens further (Figure 4.10d). The supersonic region is larger than seen in Figure 4.10c.

Lift, drag, and moment curves for $0^o \leq \alpha \leq 5.75^o$ and $M_\infty = 0.85$ are displayed for a range of α in Figure 4.11. It is interesting to note the nonlinear character of the lift curve for this transonic Mach number. C_l is relatively linear up to approximately 1.25^o and then is rather nonlinear for the rest of the range of α. The drag coefficient has a parabolic nature up to approximately $\alpha = 1.5^o$ and

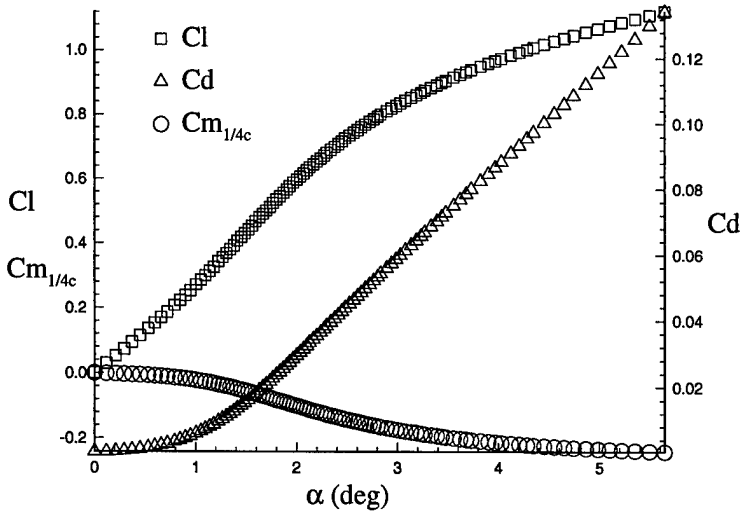


Figure 4.11 Variation of Angle-of-Attack: NACA 64A006, $M_\infty = 0.85$

then becomes fairly linear for the remainder of the range of α. The moment curve slope decreases until approximately $\alpha = 2.5^o$ and then increases for the remainder of the range of α.

V. *Airfoils With Structural Coupling*

The primary objective of the dissertation research is to develop tools to find the point at which there is a fluid-structure interaction of a dynamic system. Dynamic fluid-structure interaction between a wing and surface flow primarily consists of bending and torsional motion of the structure. A two-dimensional representation of wing bending and torsional motion is captured by a pitch and plunge airfoil (PAPA). The airfoil pitch motion simulates wing torsion, and airfoil plunge motion simulates wing bending. The coupled fluid-structure system of the current research consists of a mating of the Euler equations, discretized with a TVD scheme, and a two-degree-of-freedom pitch and plunge structural model. This combined system is a nonlinear system of equations, written as $Y_t = G(Y; \tilde{\lambda})$ and is termed the PAPA model. A description of the structural model and the various methods of solving the nonlinear system of equations is presented in this chapter.

Figure 5.1 describes the PAPA model used in the current research. Linear and torsional springs are attached to a non-deforming airfoil at the elastic axis. The elastic axis is displaced by some distance from the center-of-gravity, providing coupled pitch and plunge motion. The airfoil has an unloaded torsional spring angle of α_0, which simulates a wing root angle-of-attack. The pitch and plunge airfoil is placed in the freestream of a compressible fluid with velocity V_∞.

Section 5.1 provides a detailed description of the aerodynamic and structural dynamic governing equations. Section 5.2 outlines validation of the PAPA model. Section 5.3 provides equilibrium solutions for a wide range of structural and aerodynamic parameters.

5.1 Fluid-Structure Interaction System

This section provides a description of the structural coupling model. The equations of motion are presented for a two-degree-of-freedom, pitch and plunge airfoil model. Implementation of the PAPA model for both equilibrium solutions and time-integration solutions is provided.

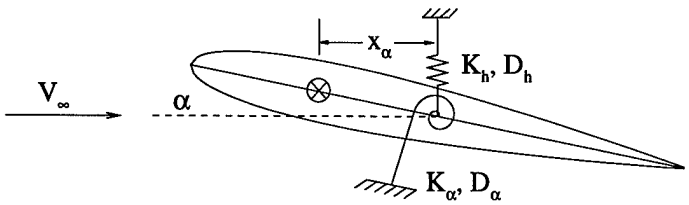


Figure 5.1 Airfoil Structural Coupling Model

5.1.1 Pitch and Plunge Structural Model. A schematic of the structural components of the PAPA model is depicted in Figure 5.1; the governing equations for the structural dynamics model are [44]

$$m\ddot{h} + mr\ddot{\alpha}\cos\alpha - mr\dot{\alpha}^2\sin\alpha + D_h\dot{h} + K_h h = L, \tag{5.1}$$

$$mr\ddot{h} + I_\alpha\ddot{\alpha} + D_\alpha\dot{\alpha} + K_\alpha(\alpha - \alpha_0) = M_{cg} + Lr\cos\alpha + Dr\sin\alpha, \tag{5.2}$$

where h is the vertical displacement, m is the airfoil mass per unit span, r is the distance from the center of gravity (cg) to the elastic axis (ea), I_α is the mass moment of inertia about the elastic axis, K_h and K_α are the spring stiffness coefficients, and D_h and D_α are the structural damping coefficients. The term α_0 defines the unloaded position of the torsional spring and models a non-zero wing root angle-of-attack (or static pretwist). L is the net applied aerodynamic force in the vertical direction, while M_{cg} is the net applied aerodynamic pitching moment about the center-of-gravity (cg).

Nondimensionalization of (5.1)-(5.2) is accomplished through the use of the following scales:

$$h^* \equiv \frac{h}{c}, \quad r^* \equiv \frac{r}{b} = x_\alpha, \quad r_\alpha^* \equiv \frac{r_\alpha}{b}, \quad t^* \equiv t\frac{V_\infty}{c}, \quad \bar{u} \equiv \frac{V_\infty}{b\omega_\alpha}, \tag{5.3}$$

where c is the chord, b is the semi-chord and the $*$ superscript implies a nondimensional quantity. The scalings chosen allow the structural equations to be compatible with the aerodynamic equations. The time scale more common in the literature is related to the pitch natural frequency, ω_α. The pitch and plunge natural frequencies and the *radius of gyration* are defined by

$$\omega_\alpha \equiv \sqrt{K_\alpha/I_\alpha}, \quad \omega_h \equiv \sqrt{K_h/m}, \quad r_\alpha \equiv \sqrt{I_\alpha/m}, \tag{5.4}$$

while the other parameters of the system are the *mass ratio*,

$$\mu_s \equiv \frac{m}{\pi\rho_\infty b^2}, \tag{5.5}$$

and the *damping ratios*,

$$\zeta_h \equiv \frac{D_h}{2\sqrt{mK_h}}, \quad \zeta_\alpha \equiv \frac{D_\alpha}{2\sqrt{I_\alpha K_\alpha}}. \tag{5.6}$$

The aerodynamic forces and moments are replaced with the coefficient of lift, C_l, the coefficient of drag, C_d, and the moment coefficient, C_m:

$$C_l \equiv \frac{L}{\rho_\infty V_\infty^2 b}, \tag{5.7}$$

$$C_d \equiv \frac{D}{\rho_\infty V_\infty^2 b}, \tag{5.8}$$

$$C_m \equiv \frac{M}{2\rho_\infty V_\infty^2 b^2}. \tag{5.9}$$

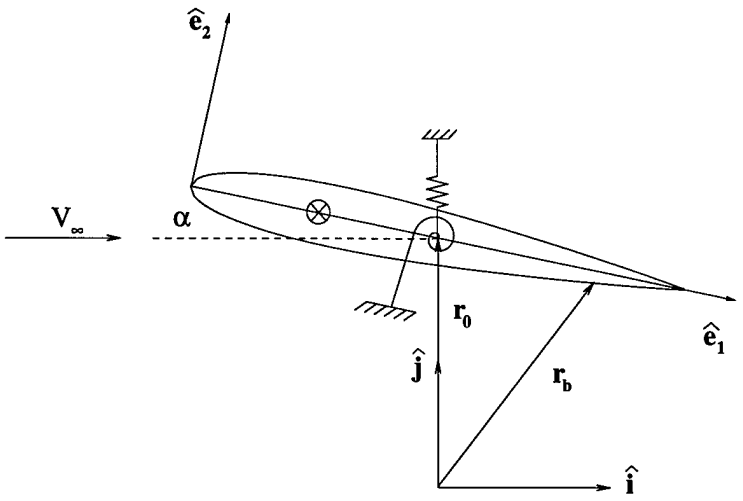


Figure 5.2 Inertial Reference Frame

Using scalings and definitions (5.3)-(5.9), and after dropping the $*$ superscript for notational convenience, (5.1)-(5.2) in nondimensional form become

$$\ddot{h} + \frac{1}{2}x_\alpha\ddot{\alpha} + 2\zeta_h\left(\frac{2}{\bar{u}}\right)\left(\frac{\omega_h}{\omega_\alpha}\right)\dot{h} + \left(\frac{2}{\bar{u}}\right)^2\left(\frac{\omega_h}{\omega_\alpha}\right)^2 h = \frac{2}{\mu_s\pi}C_l, \tag{5.10}$$

$$x_\alpha\ddot{h} + \frac{1}{2}r_\alpha^2\ddot{\alpha} + \zeta_\alpha\left(\frac{2}{\bar{u}}\right)r_\alpha^2\dot{\alpha} + \frac{1}{2}\left(\frac{2}{\bar{u}}\right)^2 r_\alpha^2(\alpha - \alpha_0) = \frac{4}{\mu_s\pi}C_{mea}, \tag{5.11}$$

where second-order terms in α are neglected and

$$C_{mea} = C_{mcg} + \frac{1}{2}x_\alpha C_l. \tag{5.12}$$

Equations (5.10)-(5.11) are solved along with the aerodynamics equations in the PAPA model.

Equations (5.10)-(5.11) are rewritten in a form compatible with a fourth-order Runge-Kutta solver. Let $s_1 = h$ and $s_3 = \alpha$, then $\dot{s}_1 = s_2$ and $\dot{s}_3 = s_4$. After substituting these definitions into

(5.10)-(5.11), the following first-order relationships are obtained:

$$\dot{s}_1 = s_2, \tag{5.13}$$

$$\dot{s}_2 + \frac{1}{2}x_\alpha \dot{s}_4 = -\left(\frac{2}{\bar{u}}\right)^2 \left(\frac{\omega_h}{\omega_\alpha}\right)^2 s_1 - 2\zeta_h \left(\frac{2}{\bar{u}}\right)\left(\frac{\omega_h}{\omega_\alpha}\right) s_2 + \frac{2}{\mu_s \pi} C_l, \tag{5.14}$$

$$\dot{s}_3 = s_4, \tag{5.15}$$

$$x_\alpha \dot{s}_2 + \frac{1}{2}r_\alpha^2 \dot{s}_4 = -\frac{1}{2}\left(\frac{2}{\bar{u}}\right)^2 r_\alpha^2 s_3 - \zeta_\alpha \left(\frac{2}{\bar{u}}\right) r_\alpha^2 s_4 + \frac{4}{\mu_s \pi} C_{mea} + \frac{1}{2}\left(\frac{2}{\bar{u}}\right)^2 r_\alpha^2 \alpha_0, \tag{5.16}$$

where the ($\dot{}$) denotes differentiation with respect to time. Rewriting (5.13)-(5.16) in matrix form yields

$$MS_t + KS = Q, \tag{5.17}$$

where

$$S = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}, \quad M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{1}{2}x_\alpha \\ 0 & 0 & 1 & 0 \\ 0 & x_\alpha & 0 & \frac{1}{2}r_\alpha^2 \end{bmatrix}, \tag{5.18}$$

$$Q = \begin{bmatrix} 0 \\ \frac{2}{\mu_s \pi} C_l \\ 0 \\ \frac{4}{\mu_s \pi} C_{mea} + \frac{1}{2}\left(\frac{2}{\bar{u}}\right)^2 r_\alpha^2 \alpha_0 \end{bmatrix}, \quad K = \begin{bmatrix} 0 & -1 & 0 & 0 \\ \left(\frac{2}{\bar{u}}\right)^2 \left(\frac{\omega_h}{\omega_\alpha}\right)^2 & 2\zeta_h \left(\frac{2}{\bar{u}}\right)\left(\frac{\omega_h}{\omega_\alpha}\right) & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & \frac{1}{2}\left(\frac{2}{\bar{u}}\right)^2 r_\alpha^2 & \zeta_\alpha \left(\frac{2}{\bar{u}}\right) r_\alpha^2 \end{bmatrix}. \tag{5.19}$$

Operating on (5.17) with M^{-1} provides the evolutionary form

$$S_t = \mathcal{K}(S, U; \tilde{\lambda}), \tag{5.20}$$

where

$$\mathcal{K} = M^{-1}Q(U) - M^{-1}KS, \tag{5.21}$$

and

$$M^{-1} = \frac{1}{\frac{1}{2}(r_\alpha^2 - x_\alpha^2)} \begin{bmatrix} \frac{1}{2}(r_\alpha^2 - x_\alpha^2) & 0 & 0 & 0 \\ 0 & \frac{1}{2}r_\alpha^2 & 0 & -\frac{1}{2}x_\alpha \\ 0 & 0 & \frac{1}{2}(r_\alpha^2 - x_\alpha^2) & 0 \\ 0 & -x_\alpha & 0 & 1 \end{bmatrix}. \tag{5.22}$$

The term Q is a function of U, since C_l and C_m are functions of U.

5.1.2 Moving Airfoil Aerodynamics Model. Modification of the equations of motion for the two-dimensional airfoil follows the time varying metric approach. The airfoil is constrained to move along the vertical inertial axis and to rotate about an elastic axis. The "O"-grid moves with the airfoil. Figure 5.2 depicts an inertial reference system with unit basis vectors $\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$ and an airfoil fixed coordinate system with unit basis vectors $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$. All of the following relationships are in nondimensional form conforming to the nondimensionalization previously discussed. The vector $\mathbf{r_0}$ is a position vector from the inertial reference frame to the elastic axis origin and is defined

$$\mathbf{r_0} = h\hat{\mathbf{j}}, \tag{5.23}$$

where h is the plunge magnitude. The local position vector $\mathbf{r_b} - \mathbf{r_0}$ is

$$\mathbf{r_b} - \mathbf{r_0} = \bar{x}\hat{\mathbf{e}}_1 + \bar{y}\hat{\mathbf{e}}_2 = x_{in}\hat{\mathbf{i}} + (y_{in} - h)\hat{\mathbf{j}}, \tag{5.24}$$

where

$$\bar{x} = x_b - x_{ea}, \quad \bar{y} = y_b - y_{ea}. \tag{5.25}$$

Herein, the subscript b refers to a point on the body, and the subscript in refers to inertial coordinates. The inertial positions, x_{in} and y_{in}, are

$$x_{in} = \bar{x}\cos\alpha + \bar{y}\sin\alpha, \quad y_{in} = -\bar{x}\sin\alpha + \bar{y}\cos\alpha + h. \tag{5.26}$$

The pitch-rate vector, $\mathbf{\Omega}$, is related to the angle of attack, α, by

$$\mathbf{\Omega} = \Omega\hat{\mathbf{k}} = -\dot{\alpha}\hat{\mathbf{k}}. \tag{5.27}$$

The inertial velocity of a point on the body in the airfoil fixed reference frame is [44]

$$\begin{aligned} \mathbf{U_b} &= \mathbf{\Omega} \times (\mathbf{r_b} - \mathbf{r_0}) + \dot{\mathbf{r}}_\mathbf{0} \\ &= (-\Omega\bar{y} - \dot{h}\sin\alpha)\hat{\mathbf{e}}_\mathbf{1} + (\Omega\bar{x} + \dot{h}\cos\alpha)\hat{\mathbf{e}}_\mathbf{2} \\ &= -\Omega(y_{in} - h)\hat{\mathbf{i}} + (\Omega x_{in} + \dot{h})\hat{\mathbf{j}}. \end{aligned} \tag{5.28}$$

The inertial acceleration of a point on the body is [44]

$$\begin{aligned} \mathbf{a_b} &= \frac{d\mathbf{\Omega}}{dt} \times (\mathbf{r_b} - \mathbf{r_0}) - \Omega^2(\mathbf{r_b} - \mathbf{r_0}) + \ddot{\mathbf{r}}_\mathbf{0} \\ &= (-\dot{\Omega}\bar{y} - \Omega^2\bar{x} - \ddot{h}\sin\alpha)\hat{\mathbf{e}}_\mathbf{1} + (\dot{\Omega}\bar{x} - \Omega^2\bar{y} + \ddot{h}\cos\alpha)\hat{\mathbf{e}}_\mathbf{2} \\ &= -(\dot{\Omega}(y_{in} - h) + \Omega^2 x_{in})\hat{\mathbf{i}} + (\dot{\Omega}x_{in} - \Omega^2(y_{in} - h) + \ddot{h})\hat{\mathbf{j}}. \end{aligned} \tag{5.29}$$

The inertial grid speed components u_g and v_g are obtained from a relationship similar to (5.28),

$$u_g = -\Omega(y_{in} - h), \quad v_g = \Omega x_{in} + \dot{h}, \tag{5.30}$$

where x_{in} and y_{in} are the inertial coordinates of the grid points. The Euler equations for a moving airfoil become

$$\frac{\partial \hat{U}}{\partial t} + \frac{\partial \hat{F}(\hat{U})}{\partial \xi} + \frac{\partial \hat{G}(\hat{U})}{\partial \eta} = 0, \tag{5.31}$$

$$\bar{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E_t \end{bmatrix}, \quad \bar{F} = \begin{bmatrix} \rho(u-u_g) \\ \rho u(u-u_g)+p \\ \rho v(u-u_g) \\ E_t(u-u_g)+pu \end{bmatrix}, \quad \bar{G} = \begin{bmatrix} \rho(v-v_g) \\ \rho u(v-v_g) \\ \rho v(v-v_g)+p \\ E_t(v-v_g)+pv \end{bmatrix}, \tag{5.32}$$

$$\hat{F} = (\xi_x \bar{F} + \xi_y \bar{G})/\mathcal{J}, \quad \hat{G} = (\eta_x \bar{F} + \eta_y \bar{G})/\mathcal{J}, \tag{5.33}$$

$$\hat{U} = \bar{U}/\mathcal{J}, \quad \mathcal{J} = \xi_x \eta_y - \xi_y \eta_x. \tag{5.34}$$

The discrete form of (5.31) for the moving airfoil is equivalent to (3.11)-(3.14):

$$L_\eta \hat{U}^n_{i,j} = \hat{U}^{n+1/2}_{i,j} = \hat{U}^n_{i,j} - \Delta t \left(\tilde{\hat{G}}^n_{i,j+\frac{1}{2}} - \tilde{\hat{G}}^n_{i,j-\frac{1}{2}} \right), \tag{5.35}$$

$$L_\xi \hat{U}^{n+1/2}_{i,j} = \hat{U}^{n+1/2}_{i,j} - \Delta t \left(\tilde{\hat{F}}^{n+1/2}_{i+\frac{1}{2},j} - \tilde{\hat{F}}^{n+1/2}_{i-\frac{1}{2},j} \right), \tag{5.36}$$

with

$$\hat{U}^{n+1}_{i,j} = L_\xi L_\eta \hat{U}^n_{i,j}. \tag{5.37}$$

Equation (5.37) is valid for

$$1 \le i \le I, \quad \text{and} \quad 2 \le j \le J-1. \tag{5.38}$$

The definitions of the TVD fluxes $\tilde{\hat{F}}$ and $\tilde{\hat{G}}$ are different for a moving airfoil than for a stationary airfoil and are defined in Appendix A.

The farfield boundary conditions for the moving airfoil are the same as (3.18)-(3.22). Also, the cut is unchanged, with (3.36) defining a mapping of variables outside the domain to physical variables inside the domain. The surface boundary conditions are different from the static airfoil

case, since a formula for impermeability on a moving surface must take into account the velocity of the surface. The normal component of velocity is the only specified condition, all other conditions are obtained from derivative conditions. The following are the surface boundary conditions similar to those defined in Subsection 3.5.3 for the static airfoil:

$$v'_s = U_{bn}, \quad \frac{\partial u'_s}{\partial n} = 0, \quad \frac{\partial \rho}{\partial n} = 0, \quad \frac{\partial E_t}{\partial n} = 0, \tag{5.39}$$

where U_{bn} is the normal velocity of the surface at the point of interest on the body. The discrete form of the surface boundary conditions are then

$$\frac{1}{2}(v'_{i,1} + v'_{i,2}) = U_{bn}, \quad u'_{i,1} = u'_{i,2}, \quad \rho_{i,1} = \rho_{i,2}, \quad E_{ti,1} = E_{ti,2}. \tag{5.40}$$

The conditions on velocity in (5.40) can be rewritten as

$$\begin{pmatrix} u'_{i,1} \\ v'_{i,1} \end{pmatrix} = \begin{pmatrix} 0 \\ 2U_{bn} \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} u'_{i,2} \\ v'_{i,2} \end{pmatrix}. \tag{5.41}$$

Following the procedures set forth in Subsection 3.5.3, the normal and tangential velocity can be written in terms of the airfoil fixed velocity components, u_e and v_e, as

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} \cos\bar{\Theta} & \sin\bar{\Theta} \\ -\sin\bar{\Theta} & \cos\bar{\Theta} \end{pmatrix} \begin{pmatrix} u_e \\ v_e \end{pmatrix}. \tag{5.42}$$

Combining (5.41) and (5.42), (5.41) becomes

$$\begin{pmatrix} \cos\bar{\Theta} & \sin\bar{\Theta} \\ -\sin\bar{\Theta} & \cos\bar{\Theta} \end{pmatrix} \begin{pmatrix} u_{ei,1} \\ v_{ei,1} \end{pmatrix} = \begin{pmatrix} 0 \\ 2U_{bn} \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \cos\bar{\Theta} & \sin\bar{\Theta} \\ -\sin\bar{\Theta} & \cos\bar{\Theta} \end{pmatrix} \begin{pmatrix} u_{ei,2} \\ v_{ei,2} \end{pmatrix}, \tag{5.43}$$

which can be rewritten as

$$\begin{pmatrix} u_{ei,1} \\ v_{ei,1} \end{pmatrix} = 2U_{bn} \begin{pmatrix} -\sin\bar{\Theta} \\ \cos\bar{\Theta} \end{pmatrix} + \begin{pmatrix} t_1(\bar{\Theta}) & t_2(\bar{\Theta}) \\ t_2(\bar{\Theta}) & -t_1(\bar{\Theta}) \end{pmatrix} \begin{pmatrix} u_{ei,2} \\ v_{ei,2} \end{pmatrix}, \tag{5.44}$$

using the definitions

$$t_1(\beta) \equiv \cos^2\beta - \sin^2\beta, \quad t_2(\beta) \equiv 2\cos\beta\sin\beta. \tag{5.45}$$

The velocity components on the surface are now expressed in the airfoil fixed reference system. To obtain the inertial velocity components, denoted by a subscript in, the following relationship is used:

$$\begin{pmatrix} u_e \\ v_e \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} u_{in} \\ v_{in} \end{pmatrix}. \tag{5.46}$$

Substituting (5.46) into (5.44), multiplying by the density, and operating on the result with the inverse of the direction cosine matrix, the variables $\rho u_{i,1}$ and $\rho v_{i,1}$ are found to be

$$\begin{aligned} \begin{pmatrix} (\rho u)_{i,1} \\ (\rho v)_{i,1} \end{pmatrix} &= 2\rho_{i,2}U_{bn} \begin{pmatrix} -\cos\alpha\sin\bar{\Theta} + \sin\alpha\cos\bar{\Theta} \\ \sin\alpha\sin\bar{\Theta} + \cos\alpha\cos\bar{\Theta} \end{pmatrix} \\ &+ \begin{pmatrix} t_1(\alpha)t_1(\bar{\Theta}) + t_2(\alpha)t_2(\bar{\Theta}) & t_1(\alpha)t_2(\bar{\Theta}) - t_2(\alpha)t_1(\bar{\Theta}) \\ t_1(\alpha)t_2(\bar{\Theta}) - t_2(\alpha)t_1(\bar{\Theta}) & -t_1(\alpha)t_1(\bar{\Theta}) - t_2(\alpha)t_2(\bar{\Theta}) \end{pmatrix} \begin{pmatrix} (\rho u)_{i,2} \\ (\rho v)_{i,2} \end{pmatrix}, \end{aligned} \tag{5.47}$$

where the boundary condition, $\rho_{i,1} = \rho_{i,2}$, appropriate for the moving mesh formulation, is used to maintain only $(i,2)$ data on the right-hand side. The normal velocity component of the body is computed from (5.28) and (5.42):

$$U_{bn} = (\Omega\bar{y} + \dot{h}\sin\alpha)\sin\bar{\Theta} + (\Omega\bar{x} + \dot{h}\cos\alpha)\cos\bar{\Theta}. \tag{5.48}$$

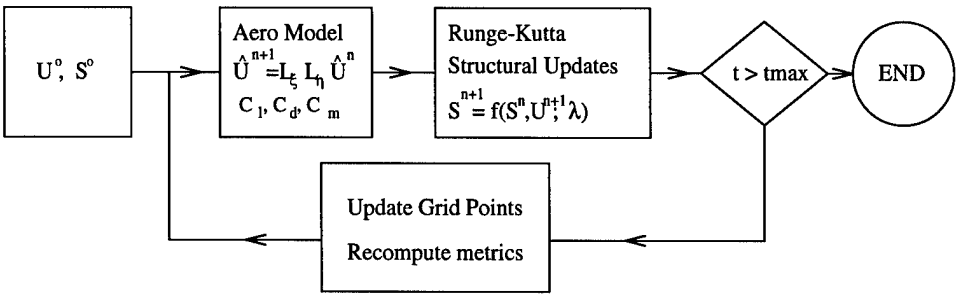


Figure 5.3 Dynamic Airfoil Algorithm Flow Chart

The two remaining conditions are

$$\rho_{i,1} = \rho_{i,2}, \quad E_{ti,1} = E_{ti,2}. \tag{5.49}$$

The surface boundary conditions for a moving airfoil consists of (5.47)-(5.49) and revert to the previously developed static airfoil model for the special case $\Omega = \dot{h} = 0$.

5.1.3 PAPA Time-Integration Algorithm. The PAPA time-integration algorithm is a loose coupling of the aerodynamics model (5.37), $\hat{U}_{i,j}^{n+1} = L_\xi L_\eta \hat{U}_{i,j}^n$, and the pitch and plunge structural model, $S_t = \mathcal{K}(S, U; \tilde{\lambda})$. Figure 5.3 depicts the flow chart of the PAPA time-integration algorithm. The flowfield solution is advanced forward in time Δt with first-order temporal accuracy using (5.37). The coefficients of lift and moment are computed from the new aerodynamic variables and then are used to advance the structural variables α, $\dot{\alpha}$, h, and $\dot{h}$ in time with a fourth-order Runge-Kutta algorithm [102]. The grid positions are then updated and the metrics recomputed. The resulting procedure has a Δt lag between the flowfield variables and the structural variables.

5.1.4 PAPA Equilibrium Algorithm. The PAPA equilibrium system is a fully implicit coupling of the aerodynamic equilibrium equations and the structural equilibrium equations. Following the approach of Section 3.7, the aerodynamic equilibrium equations are

$$F(U, S; \tilde{\lambda}) = 0, \tag{5.50}$$

and the equilibrium structural equations are defined by

$$\mathcal{K}(S, U; \tilde{\lambda}) = 0, \tag{5.51}$$

where $\mathcal{K}$ is defined in (5.21). The equilibrium fluid-structure interaction system can be rewritten in coupled form as

$$G(Y; \tilde{\lambda}) = 0, \tag{5.52}$$

where

$$Y = \begin{bmatrix} U \\ S \end{bmatrix}, \quad G = \begin{bmatrix} F \\ \mathcal{K} \end{bmatrix}. \tag{5.53}$$

The total number of equations in (5.52) is

$$N = 4I(J-1) + 4. \tag{5.54}$$

Solutions of (5.52) are calculated by Newton's method. One Newton iterate is obtained by solving the linear system

$$G_Y \Delta Y = -G. \tag{5.55}$$

The Jacobian matrix G_Y is given by

$$G_Y = \begin{bmatrix} F_U & F_S \\ \mathcal{K}_U & \mathcal{K}_S \end{bmatrix}. \tag{5.56}$$

Newton iterates are repeated until convergence is obtained (or divergence is established). The bordered algorithm is used to solve efficiently the linear system, (5.55), and is described in detail in Appendix C.

The elements of G_Y are obtained through evaluation of finite-difference approximations (Appendix B). As depicted in (5.56), the Jacobian matrix is partitioned into aerodynamic and structural blocks of elements. The structure of F_U is discussed in Chapter III for a static airfoil and maintains the bordered banded form for the moving airfoil generalization. The structure of F_S and $\mathcal{K}_U$ are four columns and four rows respectively, and $\mathcal{K}_S$ is a four-by-four matrix.

5.2 PAPA Validation

Three methods are used to validate the PAPA algorithm. Errors associated with integration of the structural equations are quantified by assuming harmonic motion in α and h, determining forcing functions, C_l and C_m, consistent with the structural equations and harmonic motion, and comparing the assumed harmonic α and h profiles with those profiles obtained through fourth-order Runge-Kutta integration. The second method of validation is a comparison of time-integration solutions for forced pitch and plunge motion obtained with the current method and an independent time-integration algorithm. The third method is a comparison of time-integration solutions for coupled, pitch and plunge motion of the PAPA model obtained with the current method, an independent time-integration algorithm, and solutions from the open literature.

Validation of the PAPA time-integration and equilibrium algorithms are presented in this section. First, the structural equations, independent of the aerodynamic equations, are validated. Next, the time-integration algorithm for a moving mesh is validated by comparison of forced motion with an independent method. Also, the PAPA time-integration algorithm is validated against two independent methods from the open literature. Finally, consistency between the equilibrium and time-integration algorithm is established. The NACA 64A006 airfoil is used for all cases. Table 5.1 defines specific run parameters and grids for each case.

Case#	*Grid#*	x_{cg}	x_α	$\zeta_h = \zeta_\alpha$	r_α^2	ω_h/ω_α	μ_s	M_∞	α_0	$\bar{u}$	Type
38	G646-4	0.5	-0.2	0	0.29	0.2	10	0.87	0	1.9	TI
39	G646-4	0.5	-0.2	0	0.29	0.2	10	0.87	0	2.0	TI
40	G646-4	0.5	-0.2	0	0.29	0.2	10	0.87	0	2.1	TI
41	G646-4	0.5	-0.2	0	0.29	0.3434	10	0.87	0	2.25	TI
42	G646-5	0.375	-0.25	1	0.25	0.2	125	0.8	1.25^o	5.0	EQ
43	G646-5	0.375	-0.25	1	0.25	0.2	125	0.8	1.25^o	5.0	TI

Table 5.1 Run Summary: PAPA Validation

5.2.1 Structural Model Validation. The PAPA structural model is validated for coupled harmonic motion in α and h. Assumed forms of $\alpha(t)$ and $h(t)$,

$$\alpha(t) = \alpha_0 + \hat{\alpha}\sin(\omega_\alpha t), \quad h(t) = \hat{h}\sin(\omega_h t), \tag{5.57}$$

are substituted into the left-hand sides of (5.10) and (5.11) to determine consistent forcing functions $C_l(t)$ and $C_m(t)$ for the right-hand sides of (5.10) and (5.11). The resulting forcing functions are

$$C_l(t) = \frac{\mu_s \pi}{2}\left[\left(-\omega_h^2 + \left(\frac{2}{\bar{u}}\right)^2\left(\frac{\omega_h}{\omega_\alpha}\right)^2\right)\hat{h}\sin\omega_h t + 2\zeta_h\left(\frac{2}{\bar{u}}\right)\left(\frac{\omega_h}{\omega_\alpha}\right)\omega_h\hat{h}\cos\omega_h t - \frac{x_\alpha}{2}\omega_\alpha^2\hat{\alpha}\sin\omega_\alpha t\right], \tag{5.58}$$

$$C_m(t) = \frac{\mu_s \pi}{4}\left[-x_\alpha\omega_h^2\hat{h}\sin\omega_h t + \frac{r_\alpha^2}{2}\left(-\omega_\alpha^2 + \left(\frac{2}{\bar{u}}\right)^2\right)\hat{\alpha}\sin\omega_\alpha t + \zeta_\alpha r_\alpha^2\left(\frac{2}{\bar{u}}\right)\omega_\alpha\hat{\alpha}\cos\omega_\alpha t\right]. \tag{5.59}$$

Integration of (5.20) is performed using the forcing functions of (5.58) and (5.59) held fixed over each time step, as in the PAPA time-integration algorithm. Histories of α and h are compared with (5.57) for various time steps to assess numerical integration accuracy.

The chosen parameters for the validation case are

$$\alpha_0 = 1.25^o, \quad \hat{\alpha} = 0.5^o, \quad \omega_\alpha = \pi, \quad \hat{h} = 0.1, \quad \omega_h = 0.2\pi,$$

$$\zeta_\alpha = \zeta_h = 0.1, \quad \bar{u} = 4, \quad \mu_s = 125, \quad x_\alpha = -0.25, \quad r_\alpha^2 = 0.25,$$

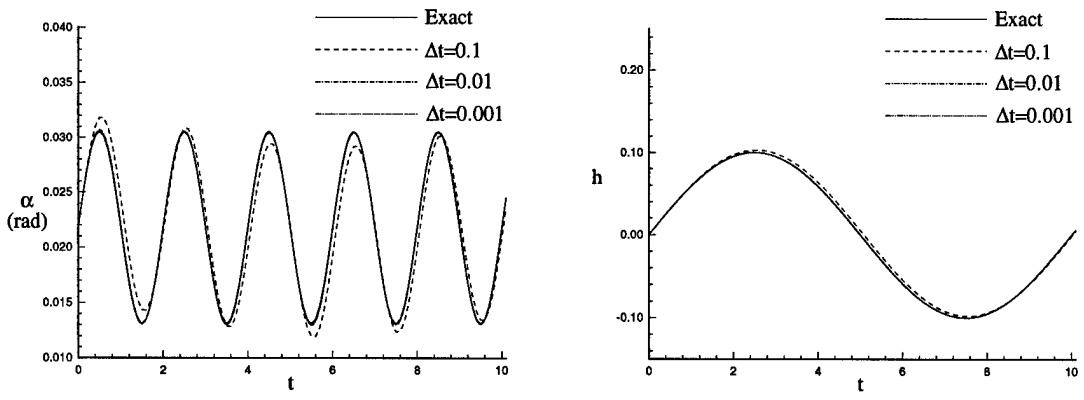


Figure 5.4 Structural Model Validation for Harmonic Motion in α and h

Δt	t_{max}	$\| \Delta\alpha \|_2$	$\| \Delta\alpha \|_\infty$	$\| \Delta h \|_2$	$\| \Delta h \|_\infty$
0.1	10	3.81×10^{-2}	8.11×10^{-3}	2.89×10^{-2}	4.84×10^{-3}
0.01	10	1.16×10^{-2}	7.63×10^{-4}	9.57×10^{-3}	5.08×10^{-4}
0.001	10	3.69×10^{-3}	7.57×10^{-5}	3.04×10^{-3}	5.10×10^{-5}
0.1	100	9.62×10^{-2}	8.11×10^{-3}	7.95×10^{-2}	5.10×10^{-3}

Table 5.2 Structural Model Integration Error Norms

which represent appropriate values for the current research. Time steps of 0.1, 0.01, and 0.001 are used to produce time histories for $0 \leq t \leq 10$.

Figures 5.4a and b are the exact solutions and numerical integrations for the chosen time steps. Table 5.2 displays maximum error norms and L_2 error norms defined as follows:

$$||\Delta\alpha|| \equiv ||\alpha - (\alpha_0 + \hat{\alpha}\sin(\omega_\alpha t))||, \quad ||\Delta h|| \equiv ||h - \hat{h}\sin(\omega_h t)||. \tag{5.60}$$

The maximum error observed in the integration of (5.20) over $0 \leq t \leq 10$, using $\Delta t = 0.01$, is 7.63×10^{-4}. This error is much smaller than errors sustained in the discretization and integration of the fluid-dynamic equations, despite maintaining a time step an order of magnitude larger than that typical of the PAPA model.

5.2.2 Forced Oscillation Validation. The dynamic airfoil algorithm is validated by Buxton [22] through a comparison of forced pitch and forced plunge motion with ENS3DAE [121], a 3-D Beam-Warming, Navier-Stokes/Euler solver. The ENS3DAE solutions are computed assuming inviscid flow. With ENS3DAE, the PAPA model is modified to treat a rectangular, rigid wing. Grid G646-1 is used as a cross-sectional two-dimensional plane of the three-dimensional grid, providing a 100 × 31 × 5 grid. Two-dimensional flow is enforced by placing inviscid planes at the wing tips. Specific parameters used in the ENS3DAE simulations as well as detailed discussions of ENS3DAE are included in [22].

A NACA 64A006 airfoil at $M_\infty = 0.87$ is forced to pitch and plunge with the following profiles:

$$\alpha(t) = \hat{\alpha}\sin(\omega_\alpha t), \quad h(t) = \hat{h}\sin(\omega_h t), \tag{5.61}$$

where

$$\omega_\alpha = 0.11, \quad \hat{\alpha} = 0.229^o, \quad \omega_h = 0.11, \quad \hat{h} = 0.028. \tag{5.62}$$

Many cycles of data are computed to ensure periodic behavior, storing one cycle of data from each method for comparison.

The time histories for forced pitch are presented in Figure 5.5 and forced plunge in Figure 5.6. The peak C_l value for the ENS3DAE forced pitch solution is 4.7% lower than the TVDntiAE peak value, whereas, the peak C_m value is 14.6% lower than the TVDntiAE solution. The peak C_l value for the ENS3DAE solution in forced plunge is 5.1% lower than the TVDntiAE peak value, whereas, the peak C_m value is 14.4% lower than the TVDntiAE solution. There are two possible sources of error: time-integration and spatial discretization/numerical dissipation model error. The spatial discretization/numerical dissipation error occurs for both a static airfoil, steady-state solution and a forced pitch and plunge solution. Time-integration errors only occur for the forced pitch and plunge solution. Since the identical grid is used (as two-dimensional planes with two-dimensional

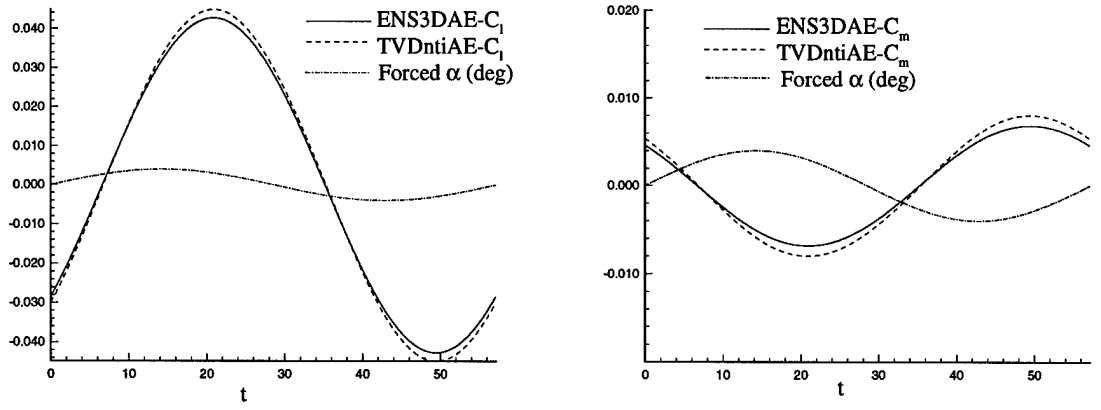


Figure 5.5 NACA 64A006 Forced Pitch Comparison: $M_\infty = 0.87$

flow verified for the ENS3DAE solution), the component of the difference between the two codes due to spatial discretization can be attributed to the difference in dissipation models, not grid resolution. ENS3DAE is a Beam-Warming algorithm with nonlinear second- and fourth-order dissipation added [121]. TVD schemes have an inherent dissipation from the upwind formulation and are able to capture shock waves accurately without overshoots, eliminating the need for adding explicit dissipation. Buxton [22] determined that for the current grid, Mach number, and with $\alpha = 1.25°$, C_l compares within 3% and C_m compares within 9% for a static airfoil calculation. This implies that time-integration must also play a part in the overall error. ENS3DAE is an Euler-implicit scheme, which is more dissipative than the TVDntiAE time-integration scheme. Both sources of error contribute to lower peak C_l and C_m values for ENS3DAE than TVDntiAE, as seen in the forced pitch and plunge solutions. In general, the two methods qualitatively compare well and quantitatively compare within 5% for C_l and 15% for C_m.

The farfield boundary conditions of TVDntiAE are cast in the form $f_t + uf_x = 0$ to allow gradient conditions to be of the same form as the interior equations. To ensure this form does not adversely effect solution quality over standard second-order extrapolation boundary conditions, comparison with a solution using second-order extrapolation is made. Both forms of TVDntiAE

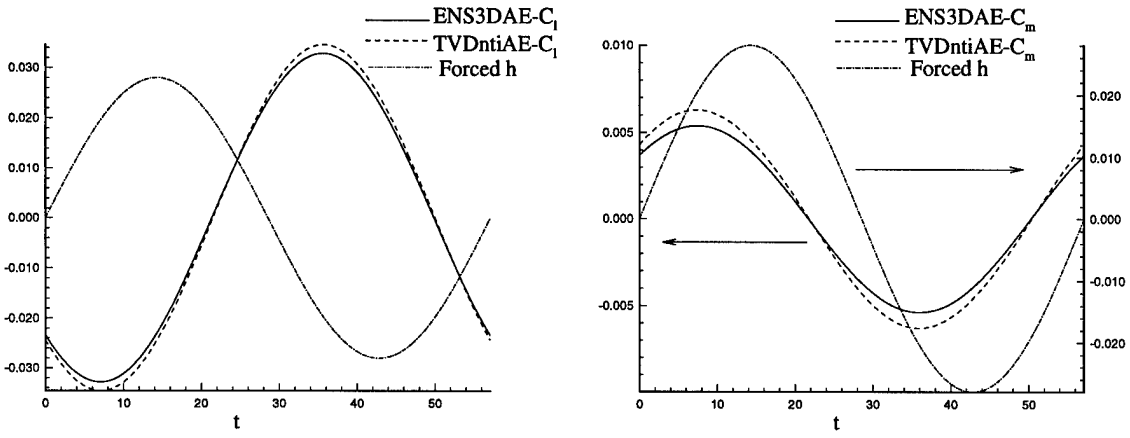


Figure 5.6 NACA 64A006 Forced Plunge Comparison: $M_\infty = 0.87$

are used to compute a forced pitch solution with the following profile and parameters:

$$\alpha(t) = \hat{\alpha}\sin(\omega_\alpha t), \tag{5.63}$$

where

$$\omega_\alpha = 0.11, \quad \hat{\alpha} = 0.229^o. \tag{5.64}$$

The resulting amplitude difference is 6.9×10^{-5} or a 0.05% difference. The phase error is 0.54 nondimensional time units or 0.4% of the period.

5.2.3 PAPA Time-Integration Algorithm Validation. Validation of the time-accurate algorithm consists of comparison with solutions reported by Kousen and Bendiksen [73] and with solutions obtained using ENS3DAE in [22], all for inviscid flow. The following is a list of the parameters used for all three methods:

$$x_{cg} = .5, \; x_\alpha = -.2, \; \zeta_h = \zeta_\alpha = 0, \tag{5.65}$$

$$\frac{\omega_h}{\omega_\alpha} = .3434, \; r_\alpha^2 = .29, \; \mu_s = 10, \; M_\infty = 0.87, \; \alpha_o = 0^o. \tag{5.66}$$

The identical structural model and coefficient of pressure, lift, and moment calculations of the current method are used in the ENS3DAE code to provide a consistent comparison. Specific numerical method parameters used in the ENS3DAE simulations and the grid are the same as those used in the previous section. The Kousen and Bendiksen (KB) method is a finite-volume, moving mesh Euler solver with added dissipation to inhibit odd-even decoupling; their grid is a "C"-grid with 96 x 16 points [73].

Figure 5.7 presents the Hopf-bifurcation diagram for the three methods. The reduced velocity associated with the flutter onset point, $\bar{u}^*$, is found to be 1.90 for the current method (time-integration), 1.92 for ENS3DAE, and 1.96 for KB, obtained through curve fits of α_{LC} for each method. Thus, the Hopf-point locations predicted by the present method and ENS3DAE differ by less than one percent, even though the dissipation models and the time-integration schemes are significantly different. The KB solution shows less than a three-percent difference in the value of $\bar{u}^*$ even though the grid, dissipation model, and time-integration scheme are very different than that used in the other two studies. The good comparison between the Hopf-point computed with the current method and with KB and ENS3DAE give confidence in the ability of the current time-integration method to compute accurately the value of $\bar{u}^*$.

Although the flutter onset point computation compares very well, the oscillation amplitude, α_{LC}, for $\bar{u} > \bar{u}^*$ differs between the three methods. In all but the largest values of $\bar{u}$, the TVDntiAE solutions are the least dissipative. Relatively large differences in α_{LC} between the present method and ENS3DAE arise from noteworthy differences (described in Section 5.2.2) in the dissipation models and time-integration schemes. However, grid or dissipation-parameter sensitivity studies to evaluate the accuracy of either method in simulating limit-cycle oscillation (LCO) have not been conducted.

5.2.4 PAPA Equilibrium and Time-Integration Algorithm Consistency. A test of the consistency between an equilibrium solution of the PAPA equilibrium algorithm and a steady-state

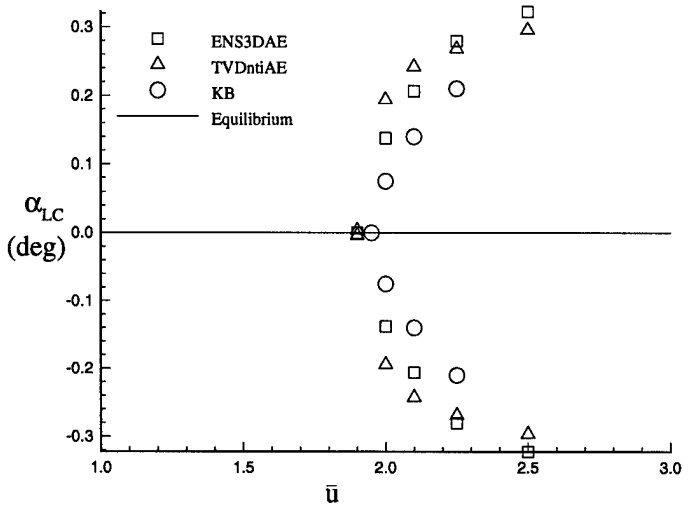


Figure 5.7 Comparison of Time-Integration Methods: $M_\infty = 0.87$, $\alpha_0 = 0^o$, and $\zeta_\alpha = \zeta_h = 0$

solution of the PAPA time-integration algorithm is provided in this section. The NACA 64A006 airfoil grid with 60 x 15 points, G646-7, is used with parameters

$$x_{cg} = .375,\ x_\alpha = -.25,\ \frac{\omega_h}{\omega_\alpha} = .2,\ r_\alpha^2 = .25,\ \mu_s = 125,\ M_\infty = 0.8,\ \alpha_o = 1.25^o. \tag{5.67}$$

The damping is chosen very large,

$$\zeta_\alpha = 1, \quad \zeta_h = 1, \tag{5.68}$$

to force the time-accurate solution to a steady-state condition in a relatively short time.

A time-accurate simulation for 190 time units is performed. Figure 5.8 depicts the time-accurate solution with equilibrium values displayed at the end time with boxes. At the end time, the time-accurate solution and the equilibrium solution differ by less than 0.02% for all four quantities. Precise agreement is necessary for the solution methods to consistently identify the Hopf-bifurcation point.

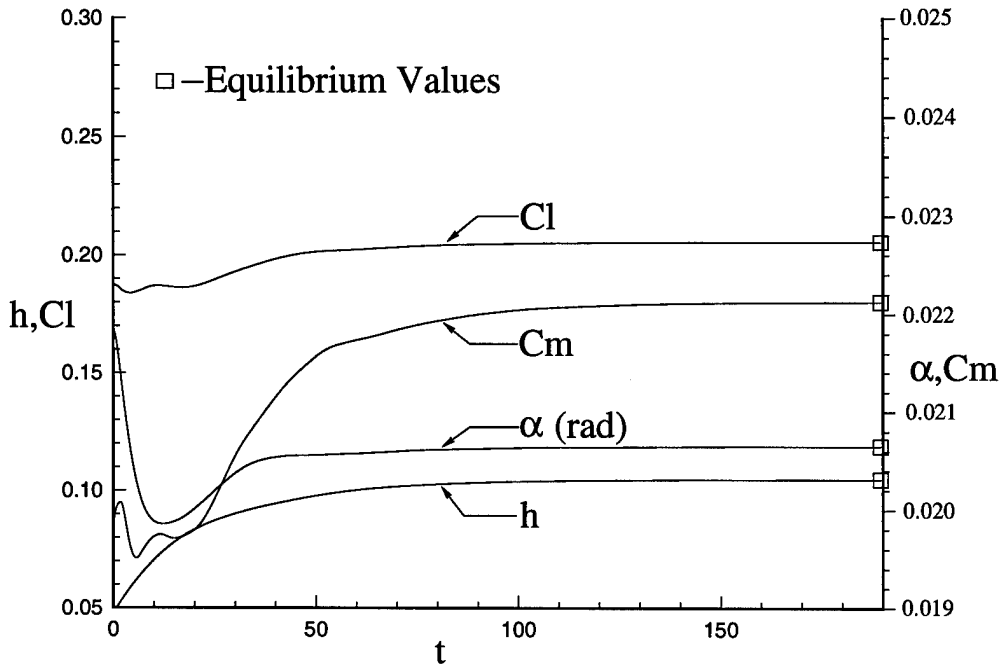


Figure 5.8 Overdamped Time Integration to Steady-State

5.3 NACA 64A006 PAPA Equilibrium Solution Space

A Hopf-point is an equilibrium solution point which satisfies additional constraints. Thus, characterizing the equilibrium solution space of a system for a wide range of parameters is important. This section presents equilibrium solutions for a NACA 64A006 PAPA model at various static pretwists, α_0, and freestream Mach numbers. The G646-2 grid is used for all cases of this section.

First, static pretwist effects are presented for a fixed Mach number, $M_\infty = 0.8$, and static pretwists in the range, $0.25^o \leq \alpha_0 \leq 1.25^o$. The solution space path for the equilibrium angle-of-attack as a function of $\bar{u}$ is presented for each α_0. Next, Mach number effects are presented for $\alpha_0 = 1.25^o$ and Mach numbers in the range, $0.63 \leq M_\infty \leq 1.2$. As in the static pretwist study, a set of equilibrium angle-of-attack paths as a function of $\bar{u}$ are presented for each Mach number. Solution space paths are efficiently generated by "manual continuation." Manual continuation is the process by which a neighboring solution is computed with the current solution as an initial guess. In this way, the number of Newton iterates necessary to obtain converged solutions is reduced by approximately 50% for the changes in $\bar{u}$ chosen for the next solution point.

5.3.1 Static Pretwist Effects. Equilibrium solutions of the PAPA model are computed for $M_\infty = 0.80$ and variations in α_0 and $\bar{u}$. Each curve of Figure 5.9 presents the equilibrium angle-of-attack, α_{eq}, versus $\bar{u}$ for a fixed, pretwist angle. For each solution computed, the airfoil pitches down from the unloaded angle of attack, $\alpha = \alpha_0$, as a result of the nose down pitching moment ($C_m < 0$). Also, the amount of pitch down increases with $\bar{u}$, since increasing $\bar{u}$ effectively weakens the torsional moment spring. As α_0 increases, the magnitude of the pitch down increases for a fixed $\bar{u}$ due to the increasingly negative pitching moment.

5.3.2 Mach Number Effects. Equilibrium solutions of the PAPA model are computed for $\alpha_0 = 1.25^o$ and variations in M_∞ and $\bar{u}$. Each curve of Figure 5.10 presents α_{eq} versus $\bar{u}$ for a fixed Mach number. Very little variation in α_{eq} is observed for $M_\infty = 0.63$. This freestream

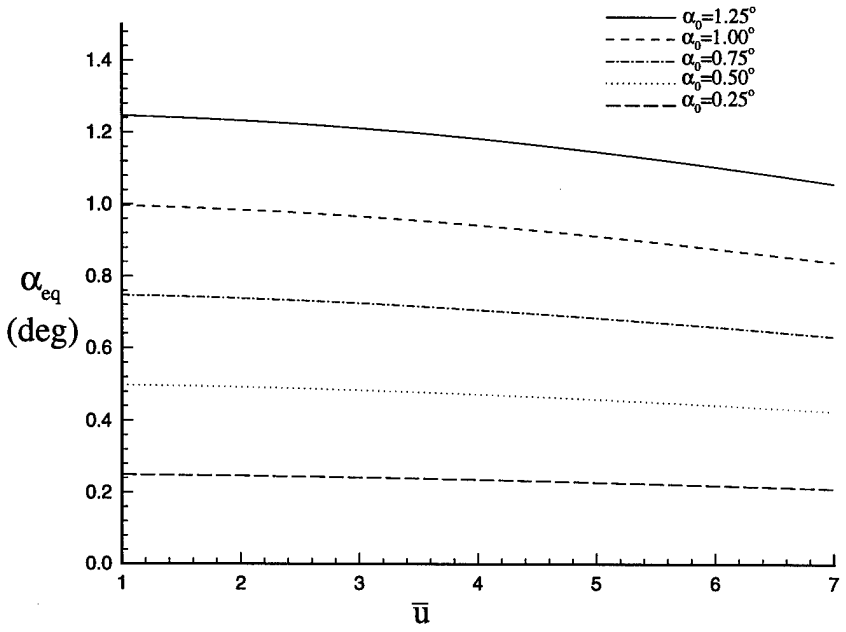


Figure 5.9 Equilibrium Angle-of-Attack: $M_\infty = 0.80$

Mach number produces subsonic flow throughout the domain for $\alpha_0 = 1.25°$. When the flowfield is completely subsonic, there is less of a negative value of C_m and the pitch down is not as severe. This is due to the relatively small expansion on the upper surface. As $\bar{u}$ increases the airfoil pitches down to a new equilibrium position. The trend of decreasing α_{eq} with increasing $\bar{u}$ is maintained for each subsonic Mach number. At a supersonic freestream Mach number of 1.2, with $\bar{u}$ fixed, the equilibrium angle-of-attack is a pitch up from the solution for $M_\infty = 0.95$. This reversal in trend is related to the phenomenon called *flutter dip* observed in transonic flow.

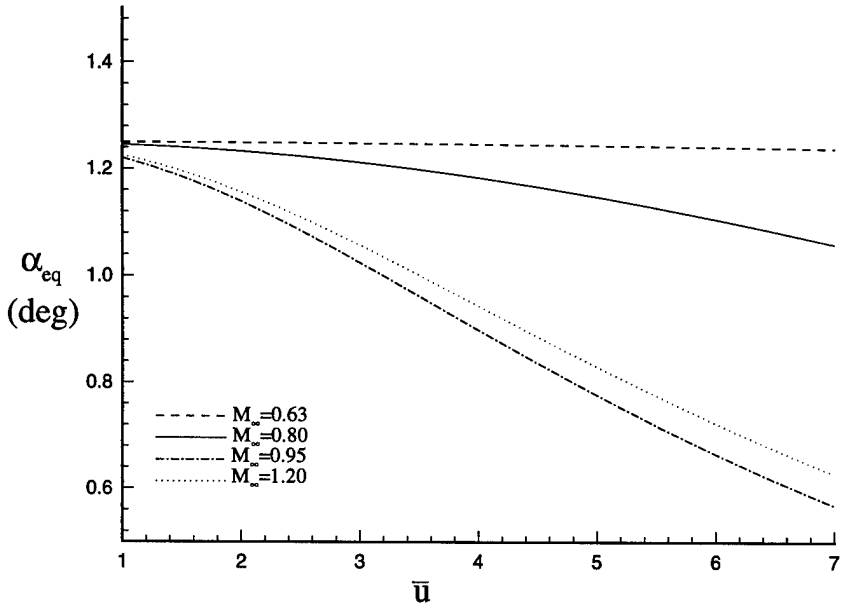


Figure 5.10 Equilibrium Angle-of-Attack: $\alpha_0 = 1.25^o$

VI. Hopf-Bifurcation Analysis

One of the main objectives of the current research is to develop a method of computing a Hopf-point with a workload on the order of a regular point, per iteration. Towards this end, a new algorithm that is more efficient when applied to a PAPA system than the Hopf-point algorithms used by previous researchers is developed.

The main algorithm used by other researchers is the Griewank and Reddien (GR) algorithm. This chapter develops modifications to the GR algorithm to eliminate forming and LU decomposing a G_Y^2 matrix, a necessary step in the GR algorithm [45]. These modifications reduce the workload to a regular point calculation with multiple right-hand sides in the solution of the linear system with LU decomposition, assuming that the LU decomposition requires the majority of computational work.

First, the Hopf-point system is defined and the structure of the system for the PAPA model is discussed. Next the GR algorithm, commonly used to solve the Hopf-point system, is described in detail and deficiencies of the algorithm for the PAPA model are discussed. Finally, an algorithm which circumvents these deficiencies is defined in detail.

6.1 General Framework for Hopf-Bifurcation Analysis

Once the nonlinear system of equations defining the fluid-structure interaction system is developed, the Hopf-point equations of Chapter II can be applied to determine the stability transition point. The system of equations, $\mathcal{F} = 0$, and the associated solution vector, $\mathcal{X}$, to be solved at the

Hopf point are written as

$$\mathcal{F} = \begin{bmatrix} G \\ G_Y P_1 + \Theta P_2 \\ G_Y P_2 - \Theta P_1 \\ q^T P_1 \\ q^T P_2 - 1 \end{bmatrix} = 0, \quad \mathcal{X} = \begin{bmatrix} Y \\ P_1 \\ P_2 \\ \tilde{\lambda} \\ \Theta \end{bmatrix}. \tag{6.1}$$

The system of equations, $\mathcal{F} = 0$, can be solved with Newton's method. The Jacobian matrix, $\mathcal{F}_\mathcal{X}$, is found by the author to be

$$\mathcal{F}_\mathcal{X} = \begin{bmatrix} G_Y & 0 & 0 & G_{\tilde{\lambda}} & 0 \\ (G_Y P_1)_Y & G_Y & \Theta I & (G_Y P_1)_{\tilde{\lambda}} & P_2 \\ (G_Y P_2)_Y & -\Theta I & G_Y & (G_Y P_2)_{\tilde{\lambda}} & -P_1 \\ 0 & q^T & 0 & 0 & 0 \\ 0 & 0 & q^T & 0 & 0 \end{bmatrix}. \tag{6.2}$$

The linear system

$$\mathcal{F}_\mathcal{X}^\nu \Delta\mathcal{X} = -\mathcal{F}^\nu \tag{6.3}$$

becomes an iterate of Newton's method for the expanded system of equations. Unfortunately, (6.3) is a $(3N + 2) \times (3N + 2)$ linear system, where N is large and defined by (5.54). This system is impractical to solve in full form for a 2-D airfoil. Algorithms that compute solutions of (6.1) for large systems approach the problem by treating $\mathcal{F}_\mathcal{X}$ as a sparse, blocked system. One such algorithm is the Griewank and Reddien [45] algorithm, the subject of the next section.

6.2 Griewank and Reddien Algorithm

The Griewank and Reddien (GR) algorithm solves the linear system (6.3) in block form to reduce the system to an $N \times N$ matrix system.

6.2.1 Development of GR Algorithm. The block method of solution is derived by examining the individual equations of (6.3):

$$G_Y \Delta Y + G_{\tilde{\lambda}} \Delta \tilde{\lambda} = N_1, \tag{6.4}$$

$$(G_Y P_1)_Y \Delta Y + G_Y \Delta P_1 + \Theta I \Delta P_2 + (G_Y P_1)_{\tilde{\lambda}} \Delta \tilde{\lambda} + P_2 \Delta \Theta = N_2, \tag{6.5}$$

$$(G_Y P_2)_Y \Delta Y - \Theta I \Delta P_1 + G_Y \Delta P_2 + (G_Y P_2)_{\tilde{\lambda}} \Delta \tilde{\lambda} - P_1 \Delta \Theta = N_3, \tag{6.6}$$

$$q^T \Delta P_1 = N_4, \tag{6.7}$$

$$q^T \Delta P_2 = N_5, \tag{6.8}$$

where N_1-N_5 are the negatives of the components of $\mathcal{F}$ in (6.1). It is assumed throughout that G_Y is nonsingular. By defining two vectors, γ_1 and γ_2, such that

$$\gamma_1 \equiv G_Y^{-1} N_1, \quad \gamma_2 \equiv G_Y^{-1} G_{\tilde{\lambda}}, \tag{6.9}$$

(6.4) becomes

$$\Delta Y = \gamma_1 - \gamma_2 \Delta \tilde{\lambda}. \tag{6.10}$$

Using (6.10), ΔY can be eliminated from (6.5) and (6.6) to provide

$$\begin{bmatrix} M_1 & M_2 & V_1 & P_2 \\ -M_2 & M_1 & V_2 & -P_1 \\ q^T & 0 & 0 & 0 \\ 0 & q^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta P_1 \\ \Delta P_2 \\ \Delta\tilde{\lambda} \\ \Delta\Theta \end{bmatrix} = \begin{bmatrix} V_3 \\ V_4 \\ N_4 \\ N_5 \end{bmatrix}, \tag{6.11}$$

where $M_1 = G_Y$, $M_2 = \Theta I$ and

$$V_1 \equiv (G_Y P_1)_{\tilde{\lambda}} - (G_Y P_1)_Y \gamma_2, \tag{6.12}$$

$$V_2 \equiv (G_Y P_2)_{\tilde{\lambda}} - (G_Y P_2)_Y \gamma_2, \tag{6.13}$$

$$V_3 \equiv N_2 - (G_Y P_1)_Y \gamma_1, \tag{6.14}$$

$$V_4 \equiv N_3 - (G_Y P_2)_Y \gamma_1. \tag{6.15}$$

The system of equations (6.11) is still very large and is not banded. To take advantage of the internally banded structure of G_Y, the process of eliminating variables is continued. First, let $M_3 \equiv M_1 M_2^{-1} = \frac{1}{\Theta} M_1$, multiply row two of (6.11) by M_3, and then add the result to row one of (6.11). By defining $M_4 \equiv M_2 + M_3 M_1 = M_2 + \frac{1}{\Theta} M_1^2$ and operating on the resulting row with M_4^{-1}, an expression for ΔP_2 is obtained:

$$\Delta P_2 = \gamma_5 - \gamma_3 \Delta\tilde{\lambda} - \gamma_4 \Delta\Theta, \tag{6.16}$$

where

$$\gamma_3 \equiv M_4^{-1}(V_1 + M_3 V_2), \quad \gamma_4 \equiv M_4^{-1}(P_2 - M_3 P_1), \quad \gamma_5 \equiv M_4^{-1}(V_3 + M_3 V_4). \tag{6.17}$$

With (6.16) substituted into the second row of (6.11), a similar expression for ΔP_1 is obtained:

$$\Delta P_1 = \frac{1}{\Theta}\left[M_1(\gamma_5 - \gamma_3\tilde{\lambda} - \gamma_4\Delta\Theta) + V_2\tilde{\lambda} - P_1\Delta\Theta - V_4\right]. \tag{6.18}$$

With representations of ΔP_1 and ΔP_2 in terms of $\Delta\tilde{\lambda}$ and $\Delta\Theta$, two equations can be developed for $\Delta\tilde{\lambda}$ and $\Delta\Theta$. First, operate on (6.16) with q^T and replace $q^T\Delta P_2$ with N_5 using the fourth row of (6.11), to obtain

$$q^T\gamma_3\Delta\tilde{\lambda} + q^T\gamma_4\Delta\Theta = q^T\gamma_5 - N_5. \tag{6.19}$$

Then, operate on (6.18) with q^T and replace $q^T\Delta P_1$ with N_4 using the third row of (6.11), to obtain

$$(-q^TM_1\gamma_3 + q^TV_2)\Delta\tilde{\lambda} + (-q^TM_1\gamma_4 - q^TP_1)\Delta\Theta = q^TV_4 - q^TM_1\gamma_5 + \Theta N_4. \tag{6.20}$$

Equations (6.19) and (6.20) can be solved with Cramers rule to obtain the scalar quantities $\Delta\tilde{\lambda}$ and $\Delta\Theta$, which can then be used to determine the other unknowns.

6.2.2 Summary of GR Algorithm. In summary, one iteration of the GR procedure is

1. solve two linear systems $G_Y\gamma_1 = N_1$ and $G_Y\gamma_2 = G_{\tilde{\lambda}}$ for γ_1 and γ_2
2. compute V_1-V_4 with equations (6.12)-(6.15)
3. compute matrices M_3 and M_4: $M_3 = \frac{1}{\Theta}M_1$, $M_4 = \Theta I + \frac{1}{\Theta}M_1^2$
4. solve three linear systems, $M_4\gamma_3 = V_1 + M_3V_2$, $M_4\gamma_4 = P_2 - M_3P_1$, and $M_4\gamma_5 = V_3 + M_3V_4$, for γ_3, γ_4, and γ_5
5. solve a system of two equations, (6.19) and (6.20), for the two unknowns $\Delta\tilde{\lambda}$ and $\Delta\Theta$ using Cramers rule
6. compute ΔP_1 and ΔP_2 with equations (6.18) and (6.16)
7. compute ΔY with equation (6.10)

8. update the variables Y, P_1, P_2, and Θ

Iterates are repeated until convergence is obtained (or divergence is established).

In Chapter I, the Beran and Lutton [15] algorithm is described and the GR algorithm limitation of solving a full matrix is presented. Equation (6.11) is also valid for the GR algorithm applied to the stream-function vorticity form of the equations, but the definition of M_2 is not ΘI. The definition of M_2 for the streamfunction-vorticity form is ΘL_1, leading to $M_3 = \frac{1}{\Theta} L_1^{-1} M_1$. Further manipulations necessary to solve the system produce a square full matrix.

6.2.3 Limitations of GR Algorithm. The GR algorithm is limited by how efficiently systems of the form $G_Y x = b$ and $M_4 x = b$ are solved. If G_Y is banded, the matrix $M_4 = \Theta I + \frac{1}{\Theta} G_Y^2$ has twice the bandwidth of G_Y. The GR algorithm coupled with LU decomposition would then have an order of work defined by an LU decomposition of G_Y and M_4 or approximately 1.5 times a regular point computation. Unfortunately, with the current algorithm, G_Y has bordered elements related to the cut and structural equations that cause G_Y to be non-banded. In general, the square of a bordered matrix is a full matrix. Therefore, M_4 is full and the order of work for the GR algorithm applied to the PAPA system is that of an LU decomposition of a full $N \times N$ system, $O(N^3)$. For a reasonably fine mesh, the GR algorithm is deemed untenable.

6.3 Blocked Gauss-Seidel Newton Algorithm

An alternative approach for computing Hopf-bifurcation points has been developed by Beran and Morton [18]. The primary purpose of the alternative approach is to maintain an order-of-magnitude equivalence, in terms of computational effort per iteration, between the calculation of Hopf-points and regular points. This method is characterized by, but not identical to, the successive (under) relaxation method and is termed a Blocked Gauss-Seidel Newton method. The acronym BGSN5 is used throughout to denote the Blocked Gauss-Seidel Newton algorithm for the five equation system, (6.1). The under-relaxation parameter is denoted by ω.

To design BGSN5, the expanded system (6.1), is rewritten with a slightly different ordering as

$$\mathcal{F}^{\nu}_{\mathcal{X}} \Delta\mathcal{X} = -\mathcal{F}^{\nu}, \tag{6.21}$$

$$\mathcal{F} = \begin{bmatrix} G_Y P_1 + \Theta P_2 \\ q^T P_1 \\ G \\ G_Y P_2 - \Theta P_1 \\ q^T P_2 - 1 \end{bmatrix} = 0, \quad \mathcal{X} = \begin{bmatrix} P_1 \\ \tilde{\lambda} \\ Y \\ P_2 \\ \Theta \end{bmatrix}, \tag{6.22}$$

so that the Jacobian matrix, $\mathcal{F}_{\mathcal{X}}$, becomes

$$\mathcal{F}_{\mathcal{X}} = \begin{bmatrix} G_Y & (G_Y P_1)_{\tilde{\lambda}} & (G_Y P_1)_Y & \Theta I & P_2 \\ q^T & 0 & 0 & 0 & 0 \\ 0 & G_{\tilde{\lambda}} & G_Y & 0 & 0 \\ -\Theta I & (G_Y P_2)_{\tilde{\lambda}} & (G_Y P_2)_Y & G_Y & -P_1 \\ 0 & 0 & 0 & q^T & 0 \end{bmatrix}. \tag{6.23}$$

First, system (6.21) is cast as a 2×2 blocked system:

$$\begin{bmatrix} D_1 & \hat{U} \\ \hat{L} & D_2 \end{bmatrix} \begin{bmatrix} \Delta\mathcal{X}_1 \\ \Delta\mathcal{X}_2 \end{bmatrix} = \begin{bmatrix} \mathcal{N}_1 \\ \mathcal{N}_2 \end{bmatrix}, \tag{6.24}$$

where

$$D_1 = \begin{bmatrix} G_Y & (G_Y P_1)_{\tilde{\lambda}} & (G_Y P_1)_Y \\ q^T & 0 & 0 \\ 0 & G_{\tilde{\lambda}} & G_Y \end{bmatrix}, \quad D_2 = \begin{bmatrix} G_Y & -P_1 \\ q^T & 0 \end{bmatrix}, \tag{6.25}$$

$$\hat{L} = \begin{bmatrix} -\Theta I & (G_Y P_2)_{\tilde{\lambda}} & (G_Y P_2)_Y \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{U} = \begin{bmatrix} \Theta I & P_2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \tag{6.26}$$

$$\mathcal{N}_1 = -\begin{bmatrix} \mathcal{F}_1 \\ \mathcal{F}_2 \\ \mathcal{F}_3 \end{bmatrix}, \quad \mathcal{N}_2 = -\begin{bmatrix} \mathcal{F}_4 \\ \mathcal{F}_5 \end{bmatrix}, \quad \Delta\mathcal{X}_1 = \begin{bmatrix} \Delta P_1 \\ \Delta\tilde{\lambda} \\ \Delta Y \end{bmatrix}, \quad \Delta\mathcal{X}_2 = \begin{bmatrix} \Delta P_2 \\ \Delta\Theta \end{bmatrix}. \tag{6.27}$$

Then, straightforward successive relaxation on the 2 × 2 blocked system (6.24) is accomplished by reducing the linear operator to a blocked lower triangular matrix with the simplification $\hat{U} = 0$. However, two modifications are made to facilitate the convergence properties of the method, of which both involve keeping some terms in $\hat{U}$. First, P_2 is kept, since this term is only a column vector, and its inclusion does not affect the speed with which the blocked, nearly lower triangular matrix is solved. Second, the four diagonal terms in ΘI corresponding to the structural dynamics equations are kept. These four diagonal terms are necessary for convergence as the main-diagonal elements of the structural dynamics equations vanish without structural damping.

With these changes, the primary source of computational work per iteration is the LU decomposition of G_Y (the aerodynamics portion) under the assumption that the numerical Jacobian calculations are negligible. This is approximately the same requirement as that for Newton's method as applied to the computation of equilibrium solutions.

6.3.1 BGSN5 Algorithm Implementation. As in the GR algorithm, the BGSN5 procedure is implemented in such a way as to primarily consist of solutions to $G_Y x = b$, for multiple right hand sides. In order to accomplish this, a partitioning of the nonlinear system, $G = 0$, into aerodynamic

and structural equations is performed,

$$G = \begin{bmatrix} F \\ \mathcal{K} \end{bmatrix} = 0, \tag{6.28}$$

resulting in a partitioned Jacobian matrix,

$$G_Y = \begin{bmatrix} F_U & F_S \\ \mathcal{K}_U & \mathcal{K}_S \end{bmatrix}. \tag{6.29}$$

In each of the vectors of the method, a subscript "a" denotes the portion of the vector pertaining to the aerodynamic equations and a subscript "b" denotes the portion pertaining to the structural equations.

The third equation in (6.21) is rewritten to obtain an expression for ΔY in terms of $\Delta\tilde{\lambda}$ by operating on the equation with with G_Y^{-1}:

$$\Delta Y = \gamma_{11} - \gamma_{12}\Delta\tilde{\lambda} \tag{6.30}$$

where

$$\gamma_{11} \equiv G_Y^{-1}N_3, \quad \gamma_{12} \equiv G_Y^{-1}G_{\tilde{\lambda}}. \tag{6.31}$$

The vectors of (6.31) can be computed with a border algorithm (outlined in Appendix C), combined with LU decomposition of the banded portion of F_U. For notational convenience, the following terms are defined:

$$\gamma_{13} \equiv (G_Y P_1)_Y \gamma_{11}, \quad \gamma_{14} \equiv (G_Y P_1)_Y \gamma_{12}, \quad \gamma_{15} \equiv (G_Y P_2)_Y \gamma_{11}, \quad \gamma_{16} \equiv (G_Y P_2)_Y \gamma_{12}. \tag{6.32}$$

The computation of $(G_Y P_1)_Y$ and $(G_Y P_2)_Y$ is described in Appendix B.

After eliminating ΔY from (6.21) and partitioning the aerodynamic equations from the structural equations, the first and fourth equations of (6.21) become

$$F_U \Delta P_{1a} + F_S \Delta P_{1b} + (G_Y P_1)_{\tilde{\lambda}_a} \Delta\tilde{\lambda} + \gamma_{13a} - \gamma_{14a} \Delta\tilde{\lambda} + P_{2a} \Delta\Theta = N_{1a}, \tag{6.33}$$

$$\mathcal{K}_U \Delta P_{1a} + \mathcal{K}_S \Delta P_{1b} + (G_Y P_1)_{\tilde{\lambda}_b} \Delta\tilde{\lambda} + \gamma_{13b} - \gamma_{14b} \Delta\tilde{\lambda} + \Theta \Delta P_{2b} + P_{2b} \Delta\Theta = N_{1b}, \tag{6.34}$$

$$-\Theta \Delta P_{1a} + (G_Y P_2)_{\tilde{\lambda}_a} \Delta\tilde{\lambda} + \gamma_{15a} - \gamma_{16a} \Delta\tilde{\lambda} + F_U \Delta P_{2a} + F_S \Delta P_{2b} - P_{1a} \Delta\Theta = N_{4a}, \tag{6.35}$$

$$-\Theta \Delta P_{1b} + (G_Y P_2)_{\tilde{\lambda}_b} \Delta\tilde{\lambda} + \gamma_{15b} - \gamma_{16b} \Delta\tilde{\lambda} + \mathcal{K}_U \Delta P_{2a} + \mathcal{K}_S \Delta P_{2b} - P_{1b} \Delta\Theta = N_{4b}. \tag{6.36}$$

Equations (6.33)-(6.36) are used to obtain relationships for ΔP_1 and ΔP_2 in terms of $\Delta\tilde{\lambda}$ and $\Delta\Theta$. After operating on (6.33) with F_U^{-1}, it becomes

$$\Delta P_{1a} + E' \Delta P_{1b} + \gamma_{1a} \Delta\tilde{\lambda} + \gamma_{2a} \Delta\Theta = \gamma_{3a}, \tag{6.37}$$

where

$$E' \equiv {F_U}^{-1} F_S, \quad \gamma_{1a} \equiv {F_U}^{-1}[(G_Y P_1)_{\tilde{\lambda}_a} - \gamma_{14a}], \quad \gamma_{2a} \equiv {F_U}^{-1} P_{2a}, \quad \gamma_{3a} \equiv {F_U}^{-1}[N_{1a} - \gamma_{13a}]. \tag{6.38}$$

By pre-multiplying (6.37) with $-\mathcal{K}_U$ and adding the result to (6.34), (6.34) becomes

$$\Delta P_{1b} + \gamma_{1b} \Delta\tilde{\lambda} + \Theta L_1^{-1} \Delta P_{2b} + \gamma_{2b} \Delta\Theta = \gamma_{3b}, \tag{6.39}$$

where

$$L_1 \equiv \mathcal{K}_S - \mathcal{K}_U E', \quad \gamma_{1b} \equiv L_1^{-1}[(G_Y P_1)_{\tilde{\lambda}_b} - \gamma_{14b} - \mathcal{K}_U \gamma_{1a}], \tag{6.40}$$

$$\gamma_{2b} \equiv L_1^{-1}[P_{2b} - \mathcal{K}_U \gamma_{2a}], \quad \gamma_{3b} \equiv L_1^{-1}[N_{1b} - \gamma_{13b} - \mathcal{K}_U \gamma_{3a}]. \tag{6.41}$$

Equation (6.39) is now used to eliminate ΔP_{1b} from (6.37) to obtain

$$\Delta P_{1a} - \Theta E' L_1^{-1} \Delta P_{2b} + (\gamma_{1a} - E'\gamma_{1b})\Delta\tilde{\lambda} + (\gamma_{2a} - E'\gamma_{2b})\Delta\Theta = \gamma_{3a} - E'\gamma_{3b}. \tag{6.42}$$

To eliminate ΔP_{1a} from (6.35), operate on (6.35) with F_U^{-1} and substitute (6.42) into the result:

$$\Delta P_{2a} + F'\Delta P_{2b} + \gamma_{4a}\Delta\tilde{\lambda} + \gamma_{5a}\Delta\Theta = \gamma_{6a}, \tag{6.43}$$

where

$$F' = {F_U}^{-1}[F_S - \Theta^2 E' L_1^{-1}], \quad \gamma_{4a} = {F_U}^{-1}[(G_Y P_2)_{\tilde{\lambda}a} + \Theta\gamma_{1a} - \Theta E'\gamma_{1b} - \gamma_{16a}], \tag{6.44}$$

$$\gamma_{5a} = {F_U}^{-1}[-P_{1a} + \Theta\gamma_{2a} - \Theta E'\gamma_{2b}], \quad \gamma_{6a} = {F_U}^{-1}[N_{4a} + \Theta\gamma_{3a} - \Theta E'\gamma_{3b} - \gamma_{15a}]. \tag{6.45}$$

The terms ΔP_{1b} and ΔP_{2a} can be eliminated from (6.36) by substituting (6.42) into (6.36) and adding to (6.43), pre-multiplied by $-\mathcal{K}_U$:

$$\Delta P_{2b} + \gamma_{4b}\Delta\tilde{\lambda} + \gamma_{5b}\Delta\Theta = \gamma_{6b}, \tag{6.46}$$

where

$$L_2 \equiv \mathcal{K}_S + \Theta^2 L_1^{-1} - \mathcal{K}_U F', \quad \gamma_{4b} \equiv L_2^{-1}[(G_Y P_2)_{\tilde{\lambda}b} + \Theta\gamma_{1b} - \gamma_{16b} - \mathcal{K}_U\gamma_{4a}], \tag{6.47}$$

$$\gamma_{5b} \equiv L_2^{-1}[-P_{1b} + \Theta\gamma_{2b} - \mathcal{K}_U\gamma_{5a}], \quad \gamma_{6b} \equiv L_2^{-1}[N_{4b} + \Theta\gamma_{3b} - \gamma_{15b} - \mathcal{K}_U\gamma_{6a}]. \tag{6.48}$$

The term ΔP_{2b} is eliminated from (6.43) by substituting (6.46) into (6.43) to obtain

$$\Delta P_{2a} + [\gamma_{4a} - F'\gamma_{4b}]\Delta\tilde{\lambda} + [\gamma_{5a} - F'\gamma_{3b}]\Delta\Theta = \gamma_{6a} - F'\gamma_{6b}. \tag{6.49}$$

Similarly, (6.46) is substituted into (6.42) to obtain

$$\Delta P_{1a} + [\gamma_{1a} - E'\gamma_{1b} + \Theta E' L_1^{-1}\gamma_{4b}]\Delta\tilde{\lambda} + [\gamma_{2a} - E'\gamma_{2b} + \Theta E' L_1^{-1}\gamma_{5b}]\Delta\Theta = \gamma_{3a} - E'\gamma_{3b} + \Theta E' L_1^{-1}\gamma_{6b}. \tag{6.50}$$

Finally, by eliminating ΔP_{2b} from (6.39) in the same manner, (6.39) becomes

$$\Delta P_{1b} + [\gamma_{1b} - \Theta L_1^{-1}\gamma_{4b}]\Delta\tilde{\lambda} + [\gamma_{2b} - \Theta L_1^{-1}\gamma_{5b}]\Delta\Theta = \gamma_{3}b - \Theta L_1^{-1}\gamma_{6b}. \tag{6.51}$$

Relationships (6.46)-(6.51) are four equations for ΔP_{1a}, ΔP_{1b}, ΔP_{2a} and ΔP_{2b} in terms of $\Delta\tilde{\lambda}$ and $\Delta\Theta$. Using the other two relationships of (6.21), $q^T\Delta P_1 = N_2$ and $q^T\Delta P_2 = N_5$, two equations for the two unknowns $\Delta\tilde{\lambda}$ and $\Delta\Theta$ can be obtained. Summing (6.50) and (6.51), operating on the result with q^T and substituting $q^T\Delta P_1 = N_2$ into the result produces

$$\xi_{11}\Delta\tilde{\lambda} + \xi_{12}\Delta\Theta = \eta_1, \tag{6.52}$$

where

$$\xi_{11} = q_a^T[\gamma_{1a} - E'\gamma_{1b} + \Theta E' L_1^{-1}\gamma_{4b}] + q_b^T[\gamma_{1b} - \Theta L_1^{-1}\gamma_{4b}], \tag{6.53}$$

$$\xi_{12} = q_a^T[\gamma_{2a} - E'\gamma_{2b} + \Theta E' L_1^{-1}\gamma_{5b}] + q_b^T[\gamma_{2b} - \Theta L_1^{-1}\gamma_{5b}], \tag{6.54}$$

$$\eta_1 = q_a^T[\gamma_{3a} - E'\gamma_{3b} + \Theta E' L_1^{-1}\gamma_{6b}] + q_b^T[\gamma_{3b} - \Theta L_1^{-1}\gamma_{6b}] - N_2. \tag{6.55}$$

Similarly, summing (6.46) and (6.49), operating on the result with q^T and substituting $q^T\Delta P_2 = N_5$ into the result produces

$$\xi_{21}\Delta\tilde{\lambda} + \xi_{22}\Delta\Theta = \eta_2, \tag{6.56}$$

where

$$\xi_{21} = q_a^T[\gamma_{4a} - F'\gamma_{4b}] + q_b^T\gamma_{4b}, \tag{6.57}$$

$$\xi_{22} = q_a^T[\gamma_{5a} - F'\gamma_{3b}] + q_b^T\gamma_{5b}, \tag{6.58}$$

$$\eta_2 = q_a^T[\gamma_{6a} - F'\gamma_{6b}] + q_b^T\gamma_{6b} - N_5. \tag{6.59}$$

The two equation system, (6.52) and (6.56), with two unknowns, $\Delta\tilde{\lambda}$ and $\Delta\Theta$, is solved with Cramers rule:

$$\Delta\tilde{\lambda} = \frac{\begin{vmatrix} \eta_1 & \xi_{12} \\ \eta_2 & \xi_{22} \end{vmatrix}}{\begin{vmatrix} \xi_{11} & \xi_{12} \\ \xi_{21} & \xi_{22} \end{vmatrix}}, \quad \Delta\Theta = \frac{\begin{vmatrix} \xi_{11} & \eta_1 \\ \xi_{21} & \eta_2 \end{vmatrix}}{\begin{vmatrix} \xi_{11} & \xi_{12} \\ \xi_{21} & \xi_{22} \end{vmatrix}}. \tag{6.60}$$

6.3.2 BGSN5 Algorithm Summary. In summary, one iteration of the BGSN5 procedure is

1. solve the bordered linear systems $G_Y\gamma_{11} = N_3$ and $G_Y\gamma_{12} = G_{\tilde{\lambda}}$ with LU decomposition and the bordering algorithm
2. compute $(G_Y P_1)_Y$ and $(G_Y P_2)_Y$ and then form γ_{13}, γ_{14}, γ_{15}, and γ_{16} with (6.32)
3. compute E', γ_{1a}, γ_{2a}, and γ_{3a} using the LU decomposed form of F_U found in step 1 with (6.38)
4. invert the 4×4 matrix L_1 and compute γ_{1b}, γ_{2b}, and γ_{3b} using (6.40) and (6.41)
5. compute F', γ_{4a}, γ_{5a}, and γ_{6a} using the LU decomposed form of F_U found in step 1 with (6.44) and (6.45)
6. invert the 4×4 matrix L_2 and compute γ_{4b}, γ_{5b}, and γ_{6b} using (6.47) and (6.48)
7. compute the corrections $\Delta\tilde{\lambda}$ and $\Delta\Theta$ with (6.60)
8. compute the corrections ΔP_1 and ΔP_2 with (6.46), (6.49), (6.50) and (6.51)
9. compute the correction ΔY with (6.30)

10. update the variables Y, P_1, P_2, $\tilde{\lambda}$, and Θ with

$$Y^{\nu+1} = Y^{\nu} + \omega \Delta Y, \tag{6.61}$$

$$P_1^{\nu+1} = P_1^{\nu} + \omega \Delta P_1, \tag{6.62}$$

$$P_2^{\nu+1} = P_2^{\nu} + \omega \Delta P_2, \tag{6.63}$$

$$\tilde{\lambda}^{\nu+1} = \tilde{\lambda}^{\nu} + \omega \Delta \tilde{\lambda}, \tag{6.64}$$

$$\Theta^{\nu+1} = \Theta^{\nu} + \omega \Delta \Theta. \tag{6.65}$$

BGSN5 iterates are repeated until convergence is obtained (or divergence is established).

6.3.3 BGSN4 Algorithm Summary. A Hopf-point computation for a symmetric airfoil has a special feature when $\alpha_0 = 0$ and $\bar{u}$ is the free parameter. Since the aerodynamic equations are independent of $\bar{u}$, $G_{\tilde{\lambda}} = G_{\bar{u}}$ is nonzero only if $\mathcal{K}_{\bar{u}}$ is nonzero. When the initial conditions for h and α are $h = \alpha = 0$, $\mathcal{K} = 0$ for any $\bar{u}$ and thus $\mathcal{K}_{\bar{u}} = 0$. Therefore, if a solution to $G = 0$ is found for a single value of $\bar{u}$, $G = 0$ for any different value of $\bar{u}$. This implies that $G = 0$ is decoupled from the other four equations of (6.21). This decoupling allows the system to be computed more efficiently by eliminating the third row and third column from the Newton iterate, (6.21). The iterative method previously described is used with the modification of eliminating the third row in D_1 and $\hat{U}$, and the third column in D_1 and $\hat{L}$. This modified method is termed the four-equation Hopf-point algorithm (BGSN4).

The procedure for one BGSN4 iterate is summarized as

1. solve the regular point problem, $G(Y; \tilde{\lambda}) = 0$, with Newton's method for Y

2. solve the linear systems,

$$F_U E' = F_S, \quad F_U \gamma_{1a} = (G_Y P_1)_{\tilde{\lambda}a}, \quad F_U \gamma_{2a} = P_{2a}, \quad F_U \gamma_{3a} = N_{1a}, \tag{6.66}$$

with LU decomposition for E', γ_{1a}, γ_{2a}, and γ_{3a}

3. invert the 4 x 4 matrix $L_1 = \mathcal{K}_S - \mathcal{K}_U E'$ and compute γ_{1b}, γ_{2b}, and γ_{3b} using

$$\gamma_{1b} = L_1^{-1}[(G_Y P_1)_{\tilde{\lambda}b} - \mathcal{K}_U \gamma_{1a}], \quad \gamma_{2b} = L_1^{-1}[P_{2b} - \mathcal{K}_U \gamma_{2a}], \quad \gamma_{3b} = L_1^{-1}[N_{1b} - \mathcal{K}_U \gamma_{3a}] \tag{6.67}$$

4. solve the linear systems,

$$F_U F' = F_S - \Theta^2 E' L_1^{-1}, \quad F_U \gamma_{4a} = (G_Y P_2)_{\tilde{\lambda}a} + \Theta \gamma_{1a} - \Theta E' \gamma_{1b}, \tag{6.68}$$

$$F_U \gamma_{5a} = -P_{1a} + \Theta \gamma_{2a} - \Theta E' \gamma_{2b}, \quad F_U \gamma_{6a} = N_{4a} + \Theta \gamma_{3a} - \Theta E' \gamma_{3b} \tag{6.69}$$

with LU backsolves for F', γ_{4a}, γ_{5a}, and γ_{6a}

5. invert the 4 x 4 matrix $L_2 = \mathcal{K}_S + \Theta^2 L_1^{-1} - \mathcal{K}_U F'$ and compute γ_{4b}, γ_{5b}, and γ_{6b} using

$$\gamma_{4b} = L_2^{-1}[(G_Y P_2)_{\tilde{\lambda}b} + \Theta \gamma_{1b} - \mathcal{K}_U \gamma_{4a}], \quad \gamma_{5b} = L_2^{-1}[-P_{1b} + \Theta \gamma_{2b} - \mathcal{K}_U \gamma_{5a}], \tag{6.70}$$

$$\gamma_{6b} = L_2^{-1}[N_{4b} + \Theta \gamma_{3b} - \mathcal{K}_U \gamma_{6a}] \tag{6.71}$$

6. compute the corrections, $\Delta\tilde{\lambda}$ and $\Delta\Theta$, with (6.60)

7. compute the corrections, ΔP_1 and ΔP_2, with (6.46)-(6.51)

8. update the variables P_1, P_2, $\tilde{\lambda}$, and Θ with

$$P_1^{\nu+1} = P_1^{\nu} + \omega \Delta P_1, \tag{6.72}$$

$$P_2^{\nu+1} = P_2^{\nu} + \omega \Delta P_2, \tag{6.73}$$

$$\tilde{\lambda}^{\nu+1} = \tilde{\lambda}^{\nu} + \omega \Delta \tilde{\lambda}, \tag{6.74}$$

$$\Theta^{\nu+1} = \Theta^{\nu} + \omega \Delta \Theta. \tag{6.75}$$

BGSN4 iterates are repeated until convergence is obtained (or divergence is established).

BGSN4 is different from BGSN5 in terms of workload in two ways. The first is the elimination of one LU backsolve, which is relatively trivial. The second difference is the elimination of $(G_Y P_1)_Y$ and $(G_Y P_2)_Y$ from the method. Since they are computed numerically and involve many more computations than G_Y (see Appendix B), the workload is reduced from the BGSN5 method. The amount of workload reduction in terms of CPU time is a topic of discussion in Chapter VII in addition to an evaluation of performance relative to computations of equilibrium solutions.

VII. PAPA Results

Hopf-bifurcation analysis is applied to a NACA 64A006 PAPA model for various grids, aerodynamic parameters, and structural parameters. Validation of the Hopf algorithm is performed by comparison with two other methods and through a limited grid sensitivity study.

Hopf-points are computed for systematic changes in Mach number, pitch and plunge damping, and static pretwist to produce flutter boundaries with respect to these variables. The resulting flutter boundaries are validated with time-integration. Cases for which $\alpha_0 = 0^o$ are computed with the BGSN4 algorithm and the static pretwist boundary is computed with the BGSN5 algorithm.

7.1 Hopf-Bifurcation Validation

The Hopf-bifurcation algorithm is validated in three ways. First, the eigenvalues of G_Y are computed at various reduced velocities to determine the stability transition point (denoted *eigenvalue transition point*). The Hopf-point, computed with TVDntiAE, is compared with the eigenvalue transition point to demonstrate consistency. Next, time-integration solutions are computed for the same reduced velocities as the eigenvalue solutions, demonstrating either damped oscillatory, or stable limit-cycle solutions. These time-integration solutions document consistency with the Hopf-point solution and the eigenvalue solutions. Finally, grid resolution effects on the Hopf-point are analyzed for a small number of grids.

The Hopf-point is computed with the BGSN4 algorithm. An equilibrium solution is computed for the PAPA system at an initial reduced velocity to be used as the initial condition for the aerodynamic and structural variables. The eigenvector components, P_1 and P_2, are initialized with the relationship

$$P_1 = P_2 = \frac{1}{\sqrt{4(N-1)}}. \tag{7.1}$$

Case#	*Grid#*	x_{cg}	x_α	$\zeta_h = \zeta_\alpha$	r_α	ω_h/ω_α	μ_s	M_∞	α_0	$\bar{u}$	Type
44	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	6.5	BGSN4
45	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	6.5	Eig
46	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	7.0	Eig
47	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	7.5	Eig
48	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	8.0	Eig
49	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	8.5	Eig
50	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	6.5	TI
51	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	7.0	TI
52	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	7.5	TI
53	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	8.0	TI
54	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	8.5	TI

Table 7.1 Run Summary: Hopf-Point Validation (Eig-eigenvalue calculation, TI-time-integration, BGSN4-Hopf-point with BGSN4 algorithm, BGSN5-Hopf-point with BGSN5 algorithm)

The normalization vector, q, is formed by the relationship

$$q = \frac{P_2}{P_2^T P_2}, \tag{7.2}$$

which satisfies the second normalization condition $q^T P_2 = 1$. The imaginary component of the eigenvalue, Θ, is initialized to 0.1 and the under-relaxation parameter is set to 0.1. The solutions are considered converged when the maximum of the L_2 residual norms of $\mathcal{F}_1 - \mathcal{F}_5$ is less than 10^{-5}. With the above initial conditions and convergence criterion, the Hopf-point solution is computed in 95 iterations. The resulting critical reduced velocity, $\bar{u}^*$, is 7.050 and the imaginary component of the critical eigenvalue pair, Θ^*, is -0.1079.

Table 7.1 lists the cases for this section. The following is a list of the aerodynamic and structural parameters used for all runs:

$$x_{cg} = .375,\ x_\alpha = -.25,\ \zeta_h = \zeta_\alpha = 0.1, \tag{7.3}$$

$$\frac{\omega_h}{\omega_\alpha} = .2,\ r_\alpha^2 = .25,\ \mu_s = 125,\ M_\infty = 0.85,\ \alpha_o = 0^o. \tag{7.4}$$

7.1.1 PAPA Eigenvalue Analysis. A demonstration of the method to compute the stability transition point with eigenvalue analysis is described in this section. The eigenvalue stability point is computed by a linear interpolation of the two eigenvalue solutions bracketing the imaginary axis. Finally, the Hopf-point computed with the BGSN4 algorithm is compared with the eigenvalue transition point.

Computing the stability transition point with eigenvalue analysis is based on the analysis of Chapter II. Time-accurate solutions close to the stability transition point can be expressed as (2.5). The time-coefficient of the exponential term determines system stability and can be computed by (2.12). Eigenvalues of G_Y, evaluated at an equilibrium point, are computed with a suite of subroutines from Numerical Recipes [102]: HQR, ELMHES, and BALANC. Eigenvalue negative real parts contribute damped modes, whereas, eigenvalues with positive real parts contribute unstable modes, which typically develop into stable limit-cycles. The eigenvalue with the largest real part is used to determine system stability. Through a variation in the reduced velocity, the stability transition point is bracketed.

Table 7.2 summarizes the eigenvalue computations for five reduced velocities obtained with cases 21-25. For cases with $\bar{u} \leq 7.0$, all eigenvalues have negative real parts and the eigenvalue with the largest real part is listed in Table 7.2. The $\bar{u} = 7.5$ and greater values in the table all have a pair of eigenvalues with positive real parts. The eigenvalue pair migration from stable to unstable as a function of $\bar{u}$ is presented in Figure 7.1a. It is evident that when $\bar{u} = 7.0$ the eigenvalue pair with largest real part is close to the imaginary axis, contributing to a very lightly damped system.

Using linear interpolation, the reduced velocity at the Hopf-point can be estimated at $\bar{u} = 7.049$ with an imaginary part $\Theta = \pm 0.10788$. The BGSN4 solution has a percent difference of 0.01% for $\bar{u}^*$ and 0.02% for Θ^*. The excellent agreement between the two methods demonstrates consistency between the eigenvalue analysis and the Hopf-point algorithm.

$\bar{u}$	Real	Imaginary
6.5	-4.0202E-03	±0.11054
7.0	-3.7365E-04	±0.10811
7.5	3.3927E-03	±0.10575
8.0	7.1845E-03	±0.10335
8.5	1.0908E-02	±0.10084

Table 7.2 Eigenvalue with Maximum Real Part for Various Reduced Velocities

7.1.2 PAPA Time-Integration Analysis. Time-integration is used to verify the oscillatory behavior of the solutions of Table 7.2. Time-accurate histories of α, h, C_l, and C_m are computed. A non-equilibrium solution ($\alpha = 0.2^o$, $h = 0$) is used as the initial condition for all but the solution for $\bar{u} = 7$. The airfoil is released from the initial conditions allowing pitch and plunge freedom. Since $\alpha = 0.2^o$ is not a stable equilibrium solution for $\alpha_0 = 0^o$, the airfoil oscillates. Due to the very light damping at $\bar{u} = 7$, this case is initialized with $\alpha = 0^o$, providing an infinitesimal perturbation in the form of truncation error.

The maximum and minimum amplitude of the angle-of-attack, α_{LC}, is recorded after a sufficient number of cycles are computed to capture the stable, oscillatory behavior (or steady-state angle-of-attack). Figure 7.1b depicts the α_{LC} versus $\bar{u}$ behavior (or Hopf curve). Solutions with $\bar{u} \leq 7$ are damped oscillatory solutions with $\alpha_{eq} = 0$. Solutions with $\bar{u} \geq 7.5$ are stable limit-cycles, for which α_{LC} increases with $\bar{u}$.

The imaginary component of the eigenvalue is an approximation of the period of oscillation. Since the eigenvalue approach only considers one frequency of oscillation in the linearization, one would expect the difference between the actual period and the period obtained with the critical eigenvalue to grow as $\bar{u}$ is increased past $\bar{u}^*$. Table 7.3 summarizes a comparison of periods obtained from eigenvalue analysis and time-integration. Both methods compute an increasing period as $\bar{u}$ increases. As expected, the percent difference between the approximated and actual periods grows as $\bar{u}$ increases. The period approximated with the Hopf-point is $T_H = \frac{2\pi}{0.1079} = 58.23$, which is consistent with the time-integration and eigenvalue analysis periods. The agreement in trend

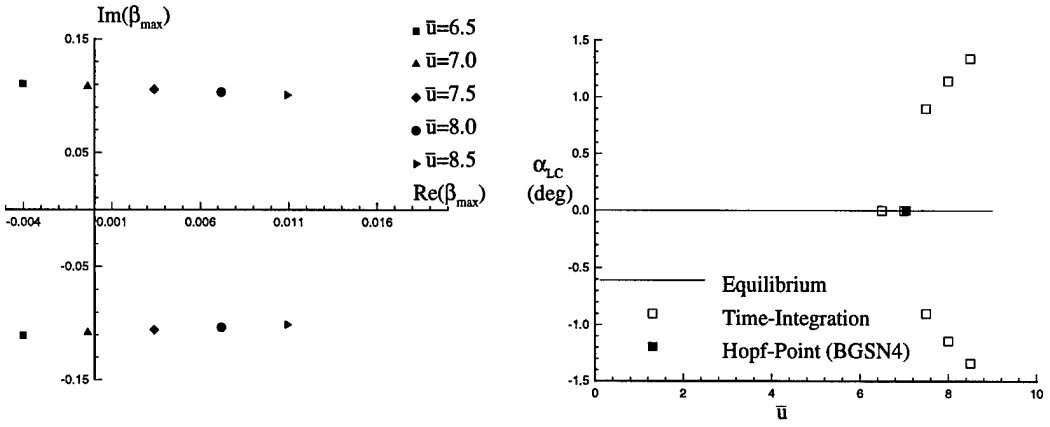


Figure 7.1 Eigenvalue Migration and Associated Hopf Curve: $\bar{u} = 6.5 - 8.5$

$\bar{u}$	T_{EIG}	T_{TI}	% Diff
7.5	59.42	56.75	4.7%
8.0	60.80	57.20	6.3%
8.5	62.31	57.33	8.7%

Table 7.3 Period of Oscillation Comparison for Various Reduced Velocities ($T_H = 58.23$ at $\bar{u}^* = 6.937$

of period and oscillatory amplitude demonstrates complete consistency between the eigenvalue analysis, time-integration solutions, and Hopf analysis.

7.1.3 Hopf-Bifurcation Point Sensitivity to Grid Resolution. The Hopf-bifurcation analysis of this work is based on a linearization of a discrete nonlinear system. The linearization incorporates an equilibrium solution and equilibrium system eigenvalues. Since equilibrium solutions and eigenvalues of discrete systems are functions of spatial discretization, it is reasonable to assume that the Hopf-point obtained with the current method is also a function of spatial discretization. This section discusses the trend in Hopf-point solutions for variations in four grid parameters, I, J, $\Delta wall$, and R_{max}. These four parameters are examined in detail in Subsection 4.1.1 to determine their effect on static airfoil solutions. In this section, variations in these four parameters is analyzed to determine their effect on the Hopf-bifurcation point. A limited number of Hopf-point solutions are compared,due to resource limitations, and trends for the critical parameters are documented.

Case#	*Grid#*	I	J	$\Delta wall$	R_{max}	$\bar{u}^*$	Θ^*	% Diff $\bar{u}^*$	% Diff Θ^*
55	G646-9	60	13	0.005	4.1	6.434	-0.1194	−7.3%	8.8%
56	G646-7	60	15	0.005	8.0	7.050	-0.1079	1.6%	-1.6%
57	G646-8	60	15	0.005	10.0	7.588	-0.1048	9.4%	-4.5%
58	G646-6	60	15	0.015	8.0	7.720	-0.1055	11.3%	-3.8%
59	G646-5	100	15	0.005	10.0	6.674	-0.1123	−3.8%	2.4%
60	G646-3	100	15	0.015	10.0	6.937	-0.1097	Baseline	Baseline

Table 7.4 Run Summary: Grid Sensitivity

Table 7.4 summarizes the cases run for the grid study. Case 60 is used as the baseline in comparing $\bar{u}^*$ and is used for the majority of results presented in later sections. The largest percent difference is 11% and encompasses variations in R_{max}, I, and resolution normal to the surface.

The number of nodes defining the airfoil shape, I, is increased from 60 to 100 in Cases 58 and 59. The other grid parameters, J, $\Delta wall$, and R_{max} are held fixed. An increase of 66% in I results in a decrease in $\bar{u}^*$ of 12% and an increase in Θ^* of 7%.

The wall spacing, $\Delta wall$, is decreased from 0.015 to 0.005 in Cases 59 and 60. All other grid parameters are held fixed. The 66% decrease in $\Delta wall$ results in a 4% decrease in $\bar{u}^*$ and an increase in Θ^* of 2%.

The domain size is increased independently of the normal resolution in Cases 55 and 56. This is accomplished by holding I and $\Delta wall$ fixed and increasing both R_{max} and J. An increase in R_{max} of 100% results in an increase in $\bar{u}^*$ of 10% and a decrease in Θ^* of 10%. In Cases 56 and 57, R_{max} is increased while holding the number of nodes normal to the surface fixed. The 20% increase in R_{max} results in a 7.6% increase in $\bar{u}^*$, implying that a decreased normal resolution also increases $\bar{u}^*$. The 20% increase in R_{max} and reduced normal resolution results in a 3% decrease in Θ^*. It is unclear whether normal resolution increases or decreases Θ^* from the combined effect since the change is not very dramatic.

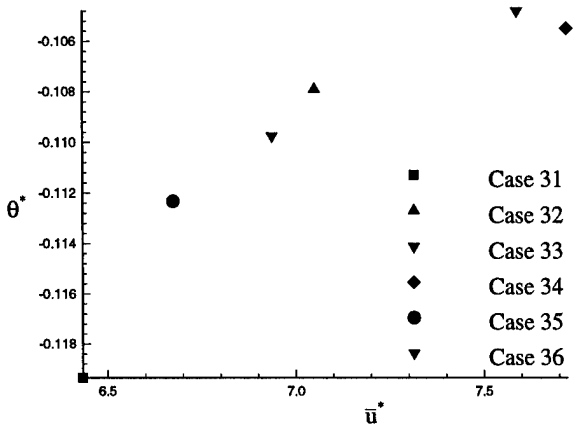


Figure 7.2 Hopf-Points for Various Grids

In summary, improving all grid parameters does not result in a consistent change in $\bar{u}^*$, as seen in Figure 7.2. Increasing the resolution around the airfoil, I, decreases $\bar{u}^*$ and increases Θ^*. Reducing $\Delta wall$ decreases $\bar{u}^*$ and increases Θ^*. Refining the grid normal to the body (i.e., increasing J), results in decreasing $\bar{u}^*$. Increasing the domain size, R_{max}, results in increasing $\bar{u}^*$ and decreasing Θ^*. The largest difference in $\bar{u}^*$ from the baseline configuration is 11% and 8.8% for Θ^*. Grid improvement has mixed effects on both $\bar{u}^*$ and Θ^*, as opposed to the static grid sensitivity study, which shows a consistent change in C_l as all four parameters are improved. Although this study is limited in scope due to resource limitations, it does provide trend information and gives evidence that the position of the Hopf-point is relatively insensitive to grid selection.

7.2 BGSN4/BGSN5 Convergence Properties

The new algorithms, BGSN4 and BGSN5, are a mixture of standard Newton's method and a blocked Gauss-Seidel, under-relaxation method. Since this hybrid method is new, it is important to document the convergence properties. This section compares the BGSN4 and BGSN5 algorithms to Newton's method, the computationally expensive alternative which is the foundation of the

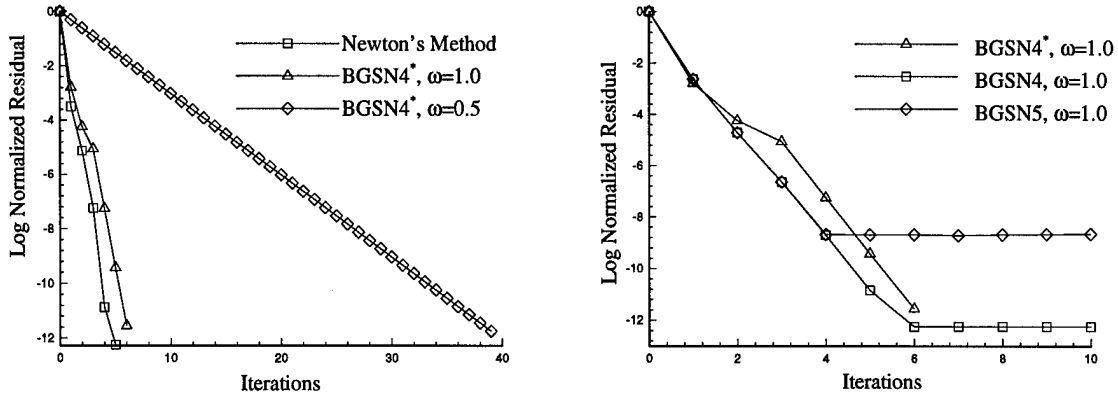


Figure 7.3 Convergence Histories for Newton's Method and Various Approximations to Newton's Method

current algorithms. Solutions for the various algorithms are computed by Beran and Morton [18] and included to describe algorithm performance.

A very coarse 30×9 grid is used to allow a full Newton's method solution of the $(3N + 2) \times (3N + 2)$ linear system. All solutions of this section are initialized with $\bar{u} = 15$, $\Theta = 0.08$, and P_1 and P_2 are initialized with (7.1). This approach defines the "theoretical best" convergence for the PAPA model at transonic Mach number conditions. The linear system (6.3) is modified to eliminate the decoupled equation, $G = 0$, from (6.1) since $\alpha_0 = 0^o$. A full matrix solver with partial pivoting is used to compute the solution. Figure 7.3a depicts the convergence history for this case. The solution converges in 5 iterations with behavior similar to the equilibrium solutions of Section 4.2.

Applying the BGSN4 modifications of $\hat{U}$ in Section 6.3 to the full matrix, a solution is computed with $\omega = 1.0$. This allows an assessment of zeroing the terms of $\hat{U}$ without the added complexity of the block manipulations implemented for efficiency. The convergence history is shown in Figure 7.3a with BGSN4* denoting the full matrix implementation. The qualitative behavior is the same as the Newton's method solution with an increase of only one iteration for the BGSN4* solution.

It is necessary to use under-relaxation parameter values less than one for grids with larger domain sizes and finer resolutions. To assess the impact of ω on convergence, values of 0.5 and 0.1 are used to compute solutions. As seen in Figure 7.3a, the character of the $\omega = 0.5$ curve is smooth and very linear. The slight "hump" observed at iteration 4 in the other two curves is non-existent in the $\omega = 0.5$ curve and the solution converges in 39 iterations. The $\omega = 0.1$ solution has the same linear behavior but takes 252 iterations to converge. The convergence history is not included in Figure 7.3a to maintain a reasonable scale on the abscissa.

The blocked-Gauss-Seidel-Newton algorithms for four and five equations are used to compute a Hopf-point solution with $\omega = 1$. The efficient blocked form of BGSN4 is used to compute a solution for direct comparison with BGSN5 (Figure 7.3b). There is a slight shifting at the third iteration in the BGSN4* solution of Figure 7.3a as compared to the BGSN4 solution of Figure 7.3b. The BGSN5 algorithm adds the complexity of an approximate $(G_Y P)_Y$ derivative on the left-hand side of the linear system. As is discussed in Appendix B, $(G_Y P)_Y$ is evaluated using a first-order-accurate, finite difference formulation, contributing to a greater degree of approximation for the Jacobian matrix of (6.3). Due to this approximated Jacobian matrix, the Newton's method implementation of (6.3) is limited to a certain level of accuracy as observed in Figure 7.3b. The first four iterations are identical for the two methods. Iterations 5-10 of the BGSN5 solution are practically unchanged in residual and solution. The BGSN4 solution continues to converge until a normalized residual of 10^{-12}, approximately four orders of magnitude of improvement. The solution, as measured by $\bar{u}^*$, is unchanged (15.0234) for both algorithms after the fourth iteration.

The numerical Jacobian parameter, ϵ_{jac}, is an important parameter in assessing convergence and accuracy of the algorithms. This is true since G_Y, a numerically computed derivative, is on both the right-hand-side and the left-hand-side of the linear system. The same value of ϵ_{jac} is used to compute both G_Y and $(G_Y P)_Y$. Figure 7.4a displays the convergence histories for BGSN5 solutions with five different numerical Jacobian parameters. The associated $\bar{u}^*$ is displayed in

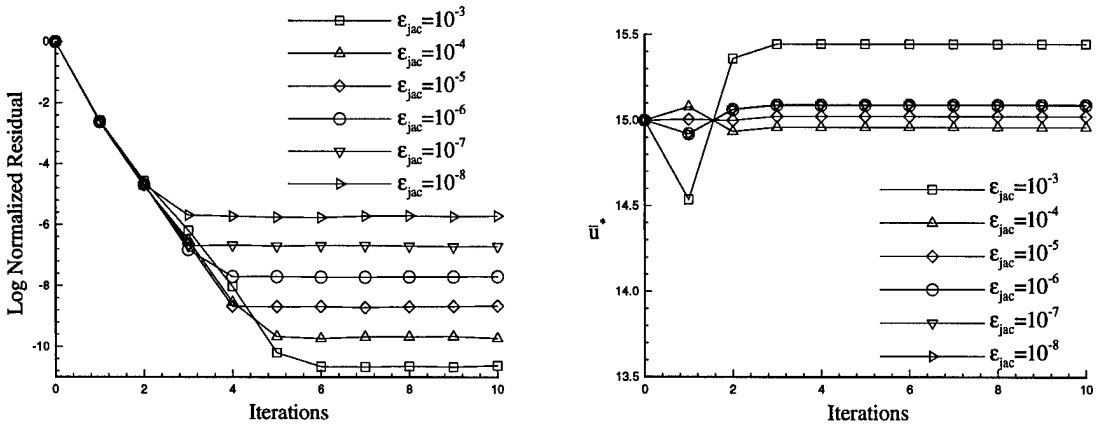


Figure 7.4 Convergence Histories for BGSN5 and BGSN4

Figure 7.4b. The larger the value of ϵ_{jac}, the lower the normalized residual, likely due to round-off error in the calculation of $(G_Y P)_Y$. Unfortunately, since G_Y is part of the nonlinear system of equations, the accuracy to which it is computed impacts not only the convergence rate but also the solution, as seen in Figure 7.4b. As the value of ϵ_{jac} is decreased, $\bar{u}^*$ converges to its asymptotic value. Solutions with $\epsilon_{jac} \leq 10^{-6}$ are virtually indistinguishable. This suggests either compromising between convergence and accuracy with the chosen ϵ_{jac} ($\epsilon_{jac} = 10^{-5}$ or 10^{-6}), or using a different ϵ_{jac} for G_Y and $(G_Y P)_Y$. The former is chosen for simplicity of coding. The difference between $\bar{u}^*$ for $\epsilon_{jac} = 10^{-5}$ and $\epsilon_{jac} = 10^{-6}$ is 0.0652 or 0.43%. Since $\epsilon_{jac} = 10^{-5}$ produces the best equilibrium solution behavior with only a slight degradation of the Hopf-point, it is chosen for the remainder of the results.

7.3 Hopf-Point Results

The Hopf-point algorithms, BGSN4 and BGSN5, are used to compute solutions for a variety of aerodynamic and structural parameters. A baseline grid is chosen and held fixed for all of the solutions of this section. First, a single Hopf-point is computed for a baseline set of parameters. Time-integration solutions are computed in the neighborhood of the Hopf-point to analyze the

Case#	*Grid#*	$\zeta_h = \zeta_\alpha$	M_∞	α_0	ω	Type	Method
61	G646-3	0.5	0.85	0	0.1	Point	BGSN4
62	G646-3	0.5	0.7-0.852	0	0.5	Boundary	BGSN4
63	G646-3	0	0.7-0.854	0	0.5	Boundary	BGSN4
64	G646-3	0-0.5	0.85	0	0.5	Boundary	BGSN4
65	G646-3	0.5	0.85	0^o-1.25^o	0.5	Boundary	BGSN4

Table 7.5 Run Summary: Hopf-Point Results

solution space. Next, the baseline solution is used to compute flutter boundaries for variations in aerodynamic and structural parameters.

A flutter boundary is modeled as a collection of Hopf-bifurcation points for different values of various parameters. Three flutter boundaries are computed. The Mach number and pitch and plunge damping flutter boundaries are computed with the BGSN4 algorithm ($\alpha_0 = 0^o$), whereas the static pretwist flutter boundary is computed with the BGSN5 algorithm. Typically, flutter boundary calculations are made with larger values of ω since a converged previous solution is used as the initial guess.

Table 7.5 is a summary of the runs in this section. The baseline grid chosen is a 100×15 grid with $\Delta wall = 0.015$ and $R_{max} = 10$. The baseline parameter set is as follows:

$$x_{cg} = .375,\ x_\alpha = -.25,\ \zeta_h = \zeta_\alpha = 0.5, \tag{7.5}$$

$$\frac{\omega_h}{\omega_\alpha} = .2,\ r_\alpha^2 = .25,\ \mu_s = 125,\ M_\infty = 0.85,\ \alpha_o = 0^o. \tag{7.6}$$

The solution is initialized as described in Section 7.1. The Hopf-point is found to be located at $\bar{u}^* = 10.2838$ with $\Theta^* = -0.1029$. Initializing $\bar{u}$ to 8, the Hopf-point is obtained after approximately 100 iterations with ω set to 0.1. Small ω values are necessary for this Mach number when initializing P_1 and P_2 as previously described.

A time-accurate simulation of the PAPA model is performed to validate the stability properties determined by the Hopf-point algorithm. The time-accurate solution is obtained by computing an

equilibrium solution at an angle-of-attack typically 0.1^o away from the equilibrium angle-of-attack and then releasing both the pitch and plunge axes. Figure 7.5a depicts the PAPA simulation for $\bar{u} = 9$. The oscillations strongly decay in the 450 time units of data presented. Figure 7.5b depicts the PAPA simulation for $\bar{u} = 10.29$, just beyond the projected Hopf-point. The oscillations are nearly neutrally stable with $\alpha_{LC} = \pm 0.0114^o$. The time histories for $\bar{u} = 11$ and $\bar{u} = 14$ are presented in Figure 7.6. The oscillations grow in amplitude until a maximum amplitude of approximately $\alpha_{LC} = \pm 0.2234^o$ for $\bar{u} = 11$ and $\alpha_{LC} = \pm 1.26^o$ for $\bar{u} = 14$. When $\bar{u} = 11$, the solution takes approximately 2000 time units to set up a stable LCO. Another method of showing the good comparison with time-integration is Figure 7.7, which displays the relationship between the computed Hopf-point and the time-integration solutions.

Near the Hopf-point, the period of oscillation can be estimated by the relationship $T_H = \frac{2\pi}{\Theta^*}$ [115]. For the Hopf-point at $\bar{u}^* = 10.2838$, $\Theta^* = 0.1029$ giving an estimated period $T_H = 61.1$. The period of oscillation in Figure 7.5b, obtained through time-integration, is 60.8, providing excellent agreement with the estimated period.

The resources required to compute a Hopf-point, a regular point, and a representative time-integration solution for two grids are shown in Table 7.6. A factor of four increase in memory is observed for a factor of two increase in the number of nodes normal to the surface. The BGSN4 solution requires a 30% increase in computational time per iteration over a regular point for the 100 x 15 grid. BGSN5, on the other hand, requires a 650% increase in computational time per iteration over a regular point for the 100 x 15 grid and a 560% increase for the 100 x 31 grid. The total number of iterations for BGSN4 and BGSN5 to converge varies as a function of ω, but for $\omega = 0.5$, a BGSN4 Hopf-point is equivalent to computing approximately four or five regular points and BGSN5 is equivalent to computing approximately 26 regular points for either grid. The six to seven fold increase in workload between BGSN4 and BGSN5 Hopf-points is due to the computations of $(G_Y P_1)_Y$ and $(G_Y P_2)_Y$ as determined by *profiling* the two computer codes.

Solution Type	Grid Size ($I \times J$)	RAM (Mb)	CPU (hrs/10 Its)	CPU (hrs/Converge)	Iterations to Converge
Regular Pt	100 × 15	33	0.51	0.51	10
BGSN4	100 × 15	47	0.66	1.93	34
BGSN5	100 × 15	67	3.84	13.2	35
Regular Pt	100 × 31	117	3.53	3.53	10
BGSN4	100 × 31	170	3.99	16.1	40*
BGSN5	100 × 31	232	23.2	93.7	40*
Representative Time-Integration Solution					
Solution Type		Grid Size ($I \times J$)		RAM (Mb)	CPU (hrs/10^6 Its)
Time-Integration		100 × 15		1.6	167.5
Time-Integration		100 × 31		2.9	509.5

Table 7.6 Computational Resource Requirements (DEC 4620/Alpha 150 Mhz Workstation). Note: $*$ denotes estimated.

Profiling is a FORTRAN method of determining how much CPU time is spent in each subroutine of the computer code. Appendix D provides a detailed description of the profiles for the solutions computed with BGSN5 and BGSN4.

Also included in Table 7.6 are run times for the time-integration algorithm. The 100 × 15 and 100 × 31 grids are used to compute dynamic solutions for 10^6 time steps. These solutions are intended to be references for performance comparison. The number of time steps chosen, although large, takes into account the number of cycles necessary to compute stable LCO solutions near Hopf-points. Table 7.6 displays the striking difference in run times between BGSN5 solutions and time-integration solutions.

Opportunities exist for improved relative performance through more efficient programming of the computational procedure for G. The efficiency can be gained by restructuring the data arrays to take advantage of the machines used to compute the current research solutions. Also, the overhead involved in computing the aerodynamic stability limit for optimal global time stepping in the time-integration algorithm is costly and can be avoided by using a constant time-step. This change of the computational procedure benefits not only the time-integration algorithm but also the equilibrium solver, due to the close coupling between the two methods. These two improvements

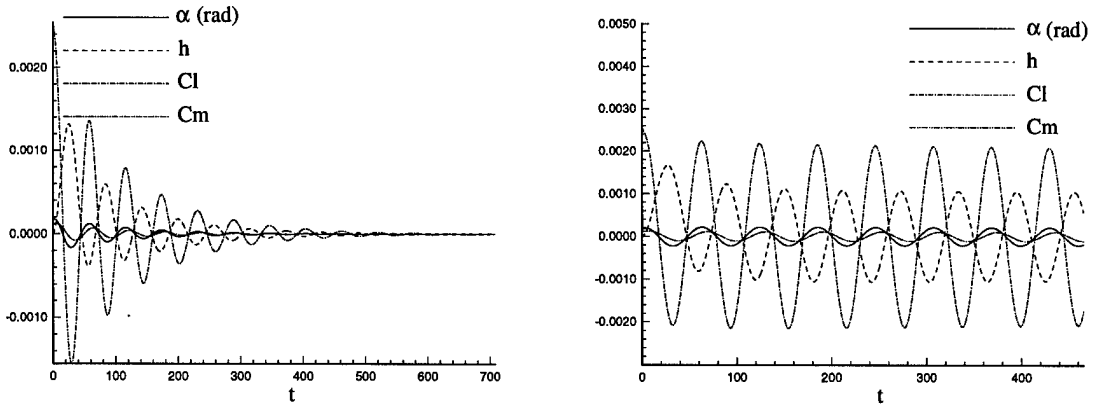


Figure 7.5 Time-Integration of PAPA Model: $\bar{u} = 9.0$ and 10.29 and $M_\infty = 0.85$

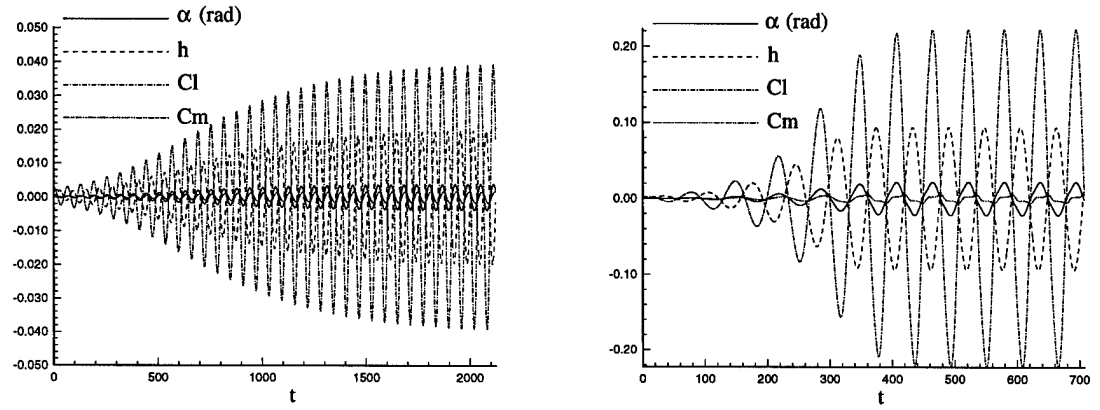


Figure 7.6 Time-Integration of PAPA Model: $\bar{u} = 11.0$ and 14 and $M_\infty = 0.85$

also impact the time-integration scheme, the function evaluation of the equilibrium method, and the numerically computed Jacobian of the equilibrium and Hopf-point algorithms.

7.3.1 Mach Flutter Boundary. The Mach flutter boundary is an important boundary to compute since it is of interest to aircraft designers. The transonic flutter dip, which is a function of airfoil shape, mass distribution, structural stiffness, and structural damping, is a limiting factor in an aircraft operational envelope. For this reason, the Mach flutter boundary of the PAPA model is computed for two pitch and plunge damping values.

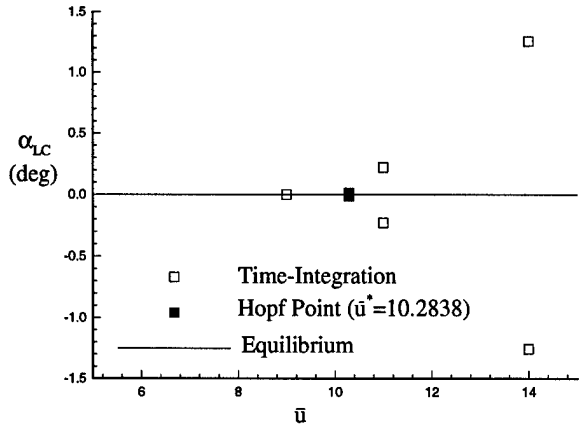


Figure 7.7 Hopf-Bifurcation Curve: $M_\infty = 0.85$, $\alpha_0 = 0^o$, and $\zeta_\alpha = \zeta_h = 0.5$

Figure 7.8 depicts the Mach flutter boundary for the baseline parameter set computed with BGSN4. Figure 7.8a presents the critical reduced velocity and Figure 7.8b presents the critical eigenvalue versus Mach number. The Hopf-point at $M_\infty = 0.85$, previously described, is used as the initial approximation for the solution at $M_\infty = 0.84$. Each additional Mach number on the flutter boundary is computed starting from the previous solution, including P_1 and P_2, until the entire boundary is computed. Every additional solution point requires approximately 20 iterations for $\omega = 0.5$, a factor of five reduction over the solution at $M_\infty = 0.85$. The complete Mach flutter boundary, comprised of 18 Hopf-points, is computed in just over 24 hours of CPU time on a DEC 4620/Alpha 150 Mhz workstation.

Time-accurate solutions near the Mach flutter boundary are computed to verify the stability boundary. Figures 7.5a and 7.6a bracket the boundary for a constant Mach number as discussed previously. Figure 7.8 provides insight into the behavior of the time-integration solution at $\bar{u} = 11$ and $M_\infty = 0.85$. Since the flutter boundary has a rapid rise near $M_\infty = 0.85$, the time-integration solution for $\bar{u} = 11$ is very close to the stability boundary and therefore is very lightly damped. The light damping contributes to the large number of oscillations necessary for a stable LCO. To verify the flutter boundary near the rapid rise, two time-integration solutions are computed for $\bar{u} = 11$

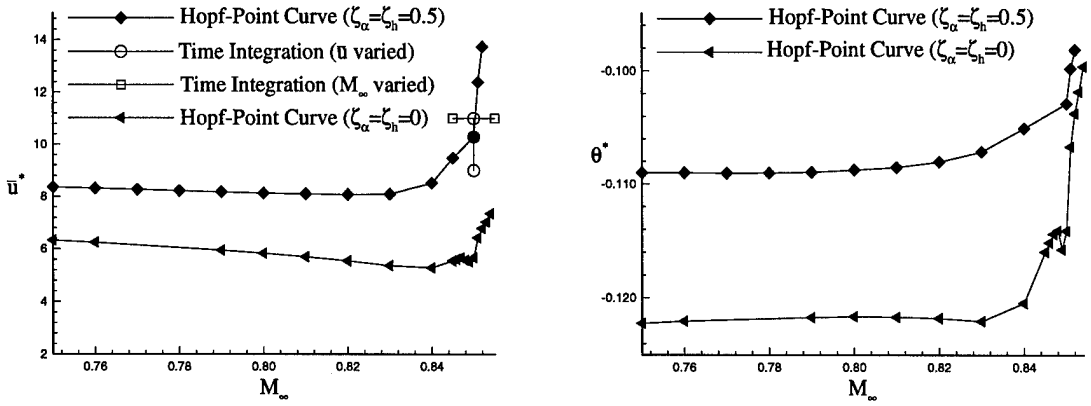


Figure 7.8 Mach Flutter Boundary: $\alpha_0 = 0^o$

and $M_\infty = 0.845$ and 0.855 (Figure 7.9). Although the Mach number variation is slight, a dramatic change in the character of the time-integration solutions is observed. The lower Mach number solution rapidly sets up on a stable limit-cycle, whereas the higher Mach number solution rapidly decays to a steady-state. This behavior is not intuitive unless the flutter boundary of Figure 7.8 is available. This highlights the usefulness of methods that compute Hopf-points directly, like the current method.

One could speculate that the sudden rise in $\bar{u}^*$ at Mach numbers near 0.85 is due to a switch in the critical eigenvalue pair, causing a dramatic difference in the character of solutions. To substantiate this speculation, additional points are computed near $M_\infty = 0.85$. Figure 7.8b depicts the change in critical eigenvalue, Θ^*, as Mach number is varied. Between $M_\infty = 0.848$ and $M_\infty = 0.849$ a sharp change in Θ^* is observed. This behavior is likely due to a single pair of eigenvalues being critical for $0.75 \leq M_\infty \leq 0.848$ and another pair critical for $M_\infty \geq 0.849$. The flutter boundary varies smoothly after $M_\infty = 0.849$ until the method is no longer able to converge (due to the lack of a sophisticated continuation method). It should be noted that to compute a curve similar to Figure 7.8b with time-integration is computationally demanding. This is due to

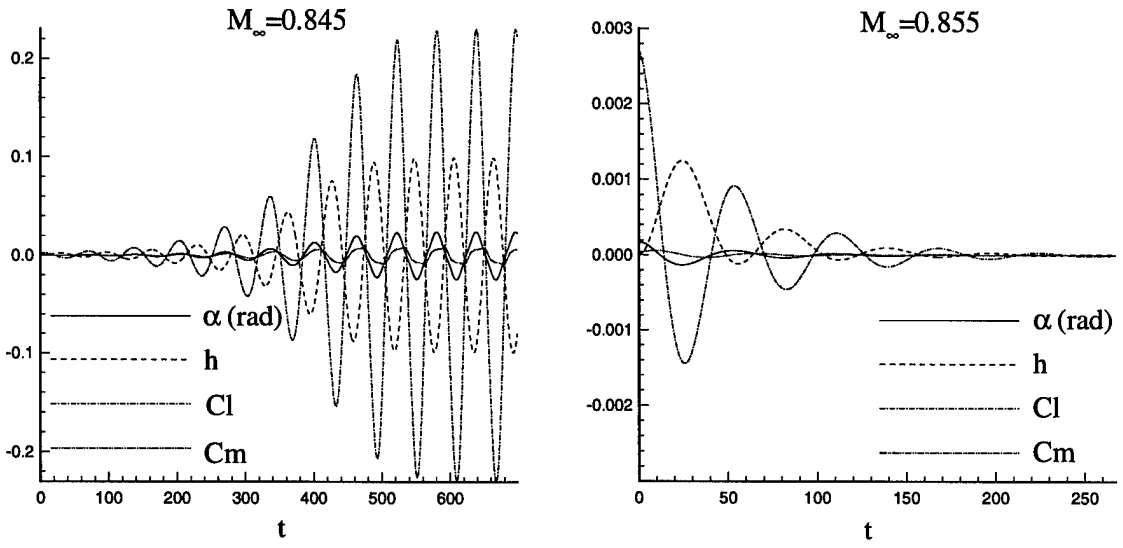


Figure 7.9 Time-Integration: $\alpha_0 = 0^o$ and $\zeta_\alpha = \zeta_h = 0.5$

the very large number of cycles required to obtain a frequency of oscillation which is not influenced by transients near a Hopf boundary.

Mach contours for one period of limit-cycle oscillation corresponding to Figure 7.6b are provided in Figure 7.10. Figure 7.11 depicts the angle-of-attack variation as a function of time and notes the location in the time-history of each solution depicted in the Mach contours. For this large $\bar{u}$ case, a lag in the shock wave motion is evident. Figure 7.10a displays the solution as the airfoil pitches up through $\alpha = 0^o$. A strong shock is evident on the lower surface and a weak shock is seen on the upper surface. The Mach contours for $\alpha = 1.26^o$ show an even stronger shock now on the upper surface and no shock is in evidence on the lower surface (Figure 7.10b). As the airfoil pitches down through $\alpha = 0^o$, the shock waves are a reflection about the chord line of the Figure 7.10a solution; the strong shock is on the upper surface and the weak shock is on the lower surface. As the airfoil reaches the minimum α (Figure 7.10d), the contours are a reflection about the chord line of the Figure 7.10b solution; the strong shock is on the lower surface and no shock is in evidence on the upper surface.

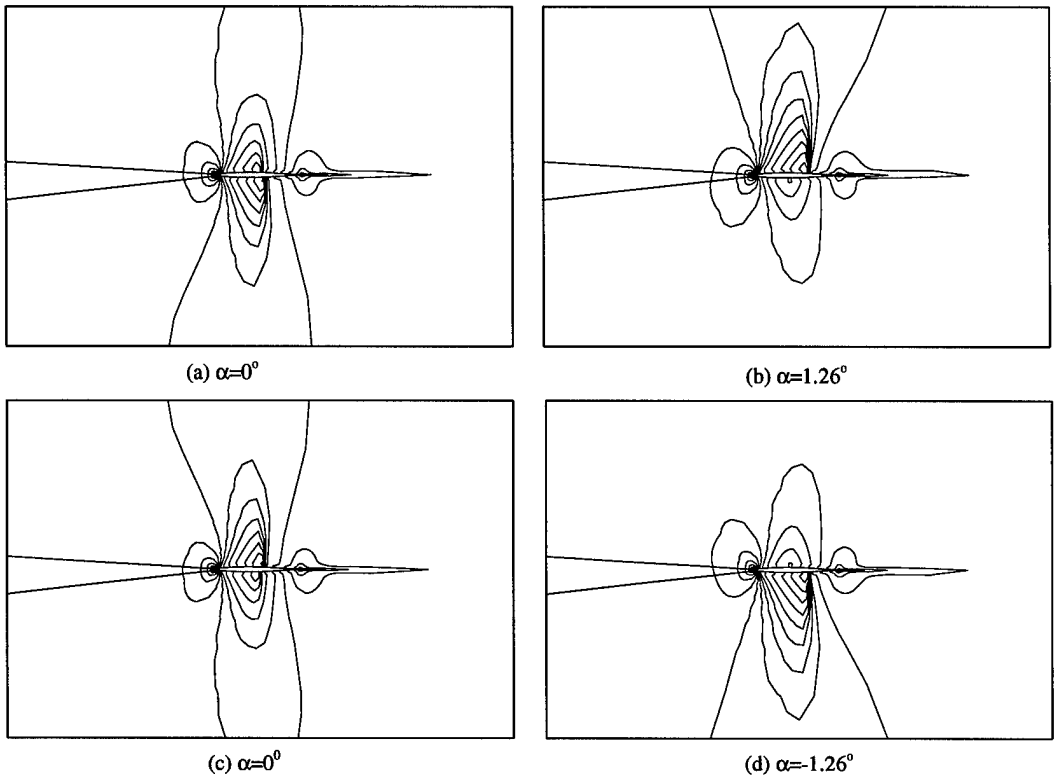


Figure 7.10 Mach Contours for a Limit-Cycle Oscillation: $M_\infty = 0.85$, $\bar{u} = 14$

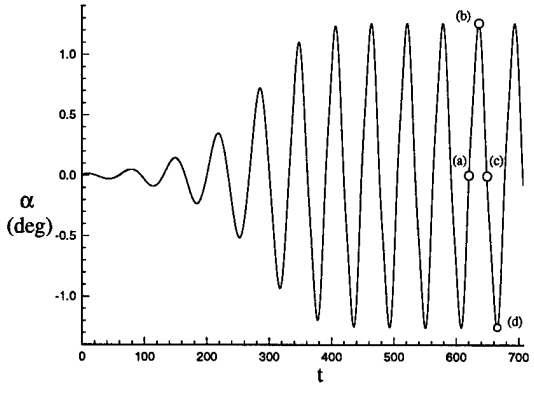


Figure 7.11 Pitch Time-History: $M_\infty = 0.85$, $\bar{u} = 14$

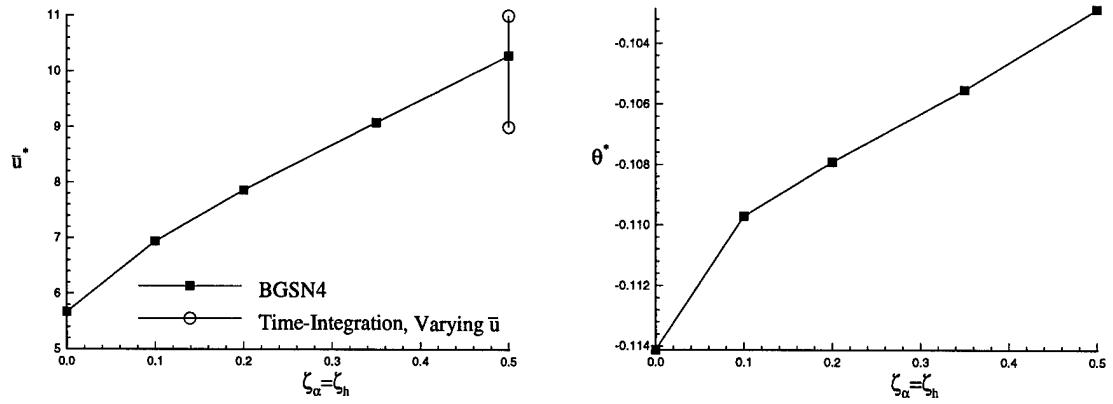


Figure 7.12 Pitch and Plunge Damping Flutter Boundary: $M_\infty = 0.85$ and $\alpha_0 = 0^o$

7.3.2 Damping Flutter Boundary. There are two primary components of the Hopf-point system: the equilibrium system, $G(Y;\tilde{\lambda}) = 0$, and the time-oscillatory system, $G_Y P = \beta P$. The damping flutter boundary is interesting to document, since variations in pitch and plunge damping have no effect on the equilibrium solution, Y^*, but do effect P_1^*, P_2^*, Θ^*, and $\bar{u}^*$. Damping also affects the numerical method. This is true, since without damping, the diagonal elements in the structural equations are all zero.

Figure 7.12 depicts the flutter boundary as pitch damping, ζ_α, and plunge damping, ζ_h, values are varied simultaneously. As depicted in Figure 7.5b and Figure 7.6a, equilibrium solutions above the flutter boundary are unstable to infinitesimal perturbations and solutions below the boundary are stable to infinitesimal perturbations. It is observed that to a high degree, $\bar{u}^*$ varies linearly with $\zeta_\alpha = \zeta_h$ over the range 0.1 to 0.5. The under-relaxation parameter necessary for convergence is affected by damping. The lower the damping, the lower ω must be for convergence. The under-relaxation parameter is set to 0.5 for pitch and plunge damping greater than or equal to 0.2 and set to 0.1 for damping less than 0.2.

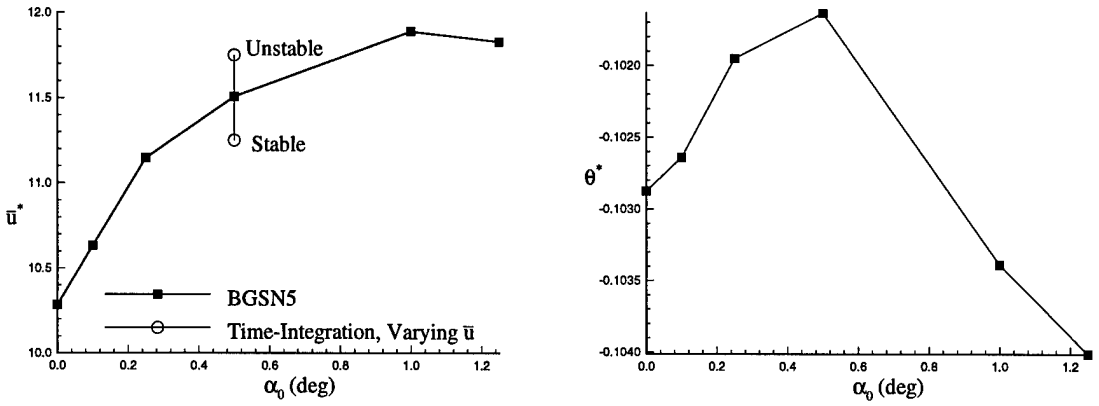


Figure 7.13 Static Pretwist Flutter Boundary: $M_\infty = 0.85$ and $\zeta_\alpha = \zeta_h = 0.5$

The behavior of Θ^* as a function of damping is displayed in Figure 7.12b. The very linear character of $\bar{u}^*$ versus $\zeta_\alpha = \zeta_h$ is reflected in the critical eigenvalue pair behavior. The linear behavior of $\bar{u}^*$ and Θ^* may be aiding in the improved convergence properties as $\zeta_\alpha = \zeta_h$ is increased.

7.3.3 Static Pretwist Flutter Boundary. The flutter boundary as a function of static pretwist is the first test of the BGSN5 algorithm for flutter boundary computation. Variations in α_0 produce changes in the equilibrium angle-of-attack and therefore changes the equilibrium aerodynamic solution. The flutter boundary then demonstrates the ability of BGSN5 to compute solutions for nontrivial, non-symmetric, aerodynamic flowfields.

The static pretwist flutter boundary is shown in Figure 7.13. Each Hopf-point on the boundary takes approximately twenty iterations to converge. The complete flutter boundary requires approximately four days of computing on a DEC 4620/ALPHA 150 Mhz workstation.

Figure 7.13b displays the behavior of Θ^* as α_0 is varied. After $\alpha_0 = 0.25^o$, there is a large change in both the reduced velocity boundary and the associated eigenvalue curve. Similar to the case of the Mach number flutter boundary, it is possible that there is a switch in the critical eigenvalue pair for α_0 in the range, $0.25^o \leq \alpha_0 \leq 0.5^o$.

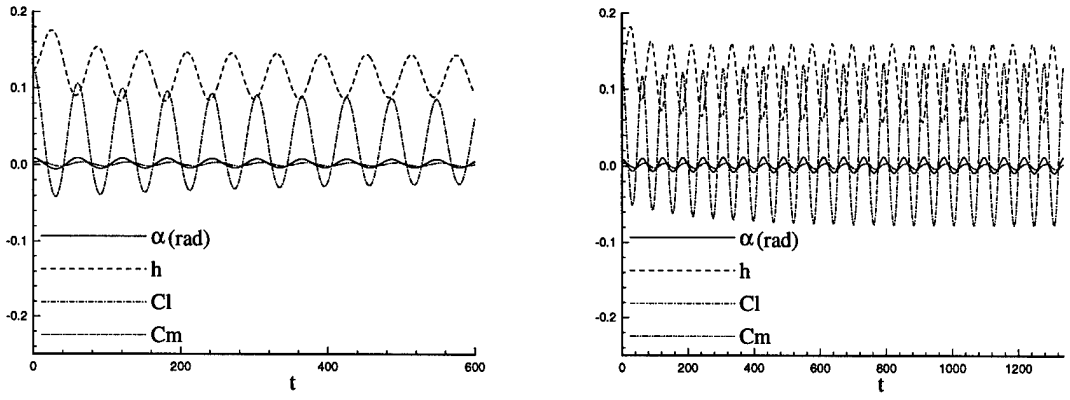


Figure 7.14 Time-Integration: $\alpha_0 = 0.5^o$, $\bar{u} = 11.25$ and $\bar{u} = 11.75$

Time-integration solutions which bracket the static pretwist flutter boundary are provided in Figure 7.14a and b. The time-integration solution for $\bar{u} = 11.25$ is lightly damped to a steady-state solution. The time-integration solution for $\bar{u} = 11.75$ is a stable limit-cycle solution after approximately 800 time units. Figure 7.14 verifies that the time-integration behavior is consistent with the predicted Hopf-point location.

VIII. Nonlinear Structural Models

Structural nonlinearities can dramatically change the location of flutter points [74]. A class of structural nonlinearities that produce bilinear torsional moments are of interest to designers. The applicability of the BGSN4 and BGSN5 algorithms to PAPA models with this class of concentrated nonlinearities is demonstrated in this chapter.

The previously described algorithms are modified to allow a class of solutions with a structural nonlinearity to be computed. The governing equations including a bilinear torsional moment are presented. Then, the structural model is validated independently of the aerodynamics model. Finally, flutter boundaries are computed for a variety of nonlinear structural model parameters.

8.1 Equations of Motion

The form of the torsional moment curve based on a bilinear structural model is depicted in Figure 8.1. The model is termed bilinear by the literature because of the two distinct slopes. The nonlinear restoring moment of the structural model, M_{rm}, is given by

$$M_{rm} = \tilde{K}_\alpha(\alpha - \alpha_0) = K_\alpha f(\alpha)(\alpha - \alpha_0), \tag{8.1}$$

where K_α is considered a global stiffness, equivalent to what is found in Chapter V. The functional form of $f(\alpha)$ is modified from the form presented in reference [20] to include a static pretwist, and takes the form

$$f(\alpha) = \begin{cases} (\alpha - \alpha_0) - \alpha_s + f_s & \text{for} \quad \alpha \geq \alpha_s + \alpha_0 \\ \frac{f_s}{\alpha_s}(\alpha - \alpha_0) & \text{for} \quad -\alpha_s + \alpha_0 < \alpha < \alpha_s + \alpha_0 \\ (\alpha - \alpha_0) + \alpha_s - f_s & \text{for} \quad \alpha \leq -\alpha_s + \alpha_0 \end{cases}, \tag{8.2}$$

or

$$f(\alpha) = f_\alpha \alpha + f_0, \tag{8.3}$$

where

$$f_\alpha = \begin{cases} 1 & \text{for } \alpha \geq \alpha_s + \alpha_0 \\ \frac{f_s}{\alpha_s} & \text{for } -\alpha_s + \alpha_0 < \alpha < \alpha_s + \alpha_0 \\ 1 & \text{for } \alpha \leq -\alpha_s + \alpha_0 \end{cases}, \tag{8.4}$$

$$f_0 = \begin{cases} -\alpha_0 - \alpha_s + f_s & \text{for } \alpha \geq \alpha_s + \alpha_0 \\ \frac{f_s}{\alpha_s}(-\alpha_0) & \text{for } -\alpha_s + \alpha_0 < \alpha < \alpha_s + \alpha_0 \\ -\alpha_0 + \alpha_s - f_s & \text{for } \alpha \leq -\alpha_s + \alpha_0 \end{cases}. \tag{8.5}$$

The pitch structural equation, (5.11), is modified to include the bilinear moment curve. Equations (5.10) and (5.11) become

$$\ddot{h} + \frac{x_\alpha}{2}\ddot{\alpha} + 2\zeta_h c_2 \dot{h} + c_2^2 h = \frac{2}{\mu_s \pi} C_l, \tag{8.6}$$

$$x_\alpha \ddot{h} + \frac{r_\alpha^2}{2}\ddot{\alpha} + \frac{2\zeta_\alpha r_\alpha^2}{\bar{u}}\dot{\alpha} + c_1 f(\alpha) = \frac{4}{\mu_s \pi} C_{mea}, \tag{8.7}$$

$$c_1 = \frac{1}{2}\left(\frac{2}{\bar{u}}\right)^2 r_\alpha^2, \quad c_2 = \left(\frac{2}{\bar{u}}\right)\left(\frac{\omega_h}{\omega_\alpha}\right). \tag{8.8}$$

The pitch natural frequency is defined in terms of the structural model for $|\alpha| \geq \alpha_s$:

$$\omega_\alpha \equiv \sqrt{K_\alpha / I_\alpha}. \tag{8.9}$$

All other parameters are defined in Chapter V.

Following the approach of Chapter V, equations (8.6)-(8.7) are rewritten in a first-order matrix form:

$$S_t = \mathcal{K}(S, U; \tilde{\lambda}), \tag{8.10}$$

where

$$\mathcal{K} = M^{-1}Q(U) - M^{-1}KS, \tag{8.11}$$

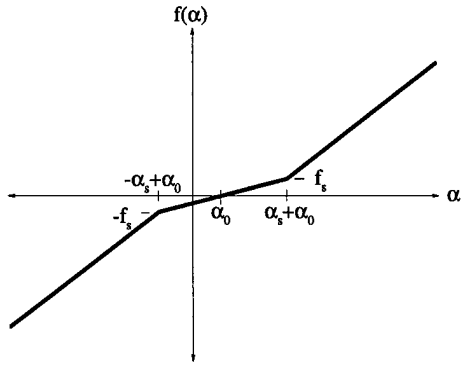


Figure 8.1 Bilinear Torsion Model

$$S = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}, \; M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{1}{2}x_\alpha \\ 0 & 0 & 1 & 0 \\ 0 & x_\alpha & 0 & \frac{1}{2}r_\alpha^2 \end{bmatrix}, \tag{8.12}$$

$$Q = \begin{bmatrix} 0 \\ \frac{2}{\mu_s \pi} C_l(U) \\ 0 \\ \frac{4}{\mu_s \pi} C_{mea}(U) - c_1 f_0 \end{bmatrix}, \quad K = \begin{bmatrix} 0 & -1 & 0 & 0 \\ c_2^2 & 2\zeta_h c_2 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & c_1 f_\alpha & \zeta_\alpha \left(\frac{2}{u}\right) r_\alpha^2 \end{bmatrix}, \tag{8.13}$$

$$M^{-1} = \frac{1}{det} \begin{bmatrix} det & 0 & 0 & 0 \\ 0 & \frac{1}{2}r_\alpha^2 & 0 & -\frac{1}{2}x_\alpha \\ 0 & 0 & det & 0 \\ 0 & -x_\alpha & 0 & 1 \end{bmatrix}, \tag{8.14}$$

$$det \equiv \frac{1}{2}(r_\alpha^2 - x_\alpha^2). \tag{8.15}$$

It should be noted that for $f_s = \alpha_s$, the model reduces to a linear structural model. Also, for $f_s = 0$ and $\alpha_s \neq 0$, a *freeplay* structural model is obtained.

8.2 Nonlinear Structural Model Validation

The PAPA structural model is validated for coupled harmonic motion in α and h following the approach of Chapter V. Assumed forms of $\alpha(t)$ and $h(t)$,

$$\alpha(t) = \alpha_0 + \hat{\alpha}\sin(\omega_\alpha t), \quad h(t) = \hat{h}\sin(\omega_h t), \tag{8.16}$$

are substituted into the left hand sides of (8.6) and (8.7) to determine consistent forcing functions $C_l(t)$ and $C_m(t)$ for the right hand sides of (8.6) and (8.7). The resulting forcing functions are

$$C_l(t) = \frac{\mu_s \pi}{2}\left[\left(-\omega_h^2 + \left(\frac{2}{\bar{u}}\right)^2\left(\frac{\omega_h}{\omega_\alpha}\right)^2\right)\hat{h}\sin\omega_h t + 2\zeta_h\left(\frac{2}{\bar{u}}\right)\left(\frac{\omega_h}{\omega_\alpha}\right)\omega_h\hat{h}\cos\omega_h t - \frac{x_\alpha}{2}\omega_\alpha^2\hat{\alpha}\sin\omega_\alpha t\right], \tag{8.17}$$

$$\begin{aligned} C_m(t) = \frac{\mu_s \pi}{4}\Big[-x_\alpha\omega_h^2\hat{h}\sin\omega_h t \;&-\; \frac{r_\alpha^2}{2}\omega_\alpha^2\hat{\alpha}\sin\omega_\alpha t + \zeta_\alpha r_\alpha^2\left(\frac{2}{\bar{u}}\right)\omega_\alpha\hat{\alpha}\cos\omega_\alpha t \\ &+\; \frac{r_\alpha^2}{2}\left(\frac{2}{\bar{u}}\right)^2\left(f_\alpha(\hat{\alpha}_0 + \hat{\alpha}\sin\omega_\alpha t) + f_0\right)\Big], \end{aligned} \tag{8.18}$$

Integration of (8.10) is performed using fourth-order Runge-Kutta integration with the forcing functions of (8.17) and (8.18). The resulting time histories are compared with (8.16) for various time steps. The forcing functions (8.17) and (8.18) are held constant over each time step in the Runge-Kutta integration to emulate the PAPA time-integration algorithm. The chosen parameters for the validation case are

$$\alpha_0 = 0^o, \;\; \hat{\alpha} = 1.5^o, \;\; \omega_\alpha = \pi, \;\; \hat{h} = 0.1, \;\; \omega_h = 0.2\pi, \;\; \alpha_s = 0.5^o$$

$$\zeta_\alpha = \zeta_h = 0.1, \;\; \bar{u} = 4, \;\; \mu_s = 125, \;\; x_\alpha = -0.25, \;\; r_\alpha^2 = 0.25.$$

These parameters are the same as the Section 5.2.1 validation with a region of secondary torsional moment slope, $-0.5 \leq \alpha \leq 0.5$, and $f_s/\alpha_s = 0.5$. Time steps of 0.1, 0.01, and 0.001 are used

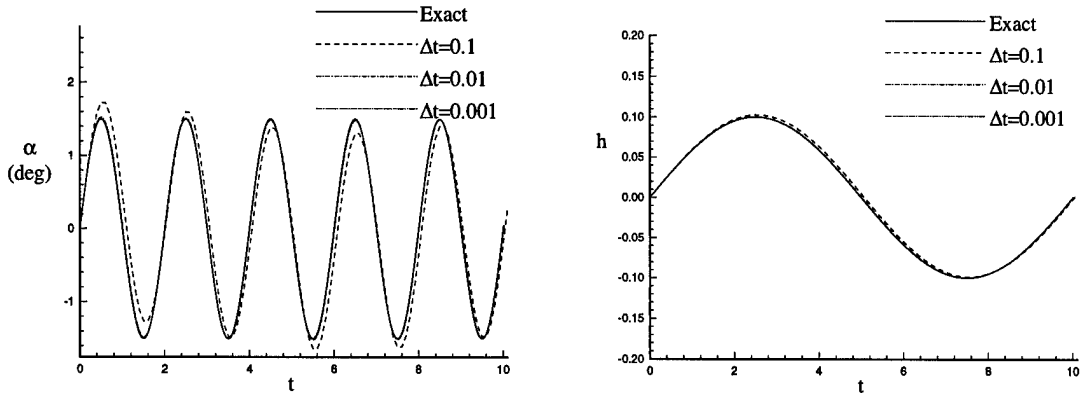


Figure 8.2 Nonlinear Structural Model Validation for Harmonic Motion in α and h

f_s/α_s	Δt	t_{max}	$\| \Delta\alpha \|_2$	$\| \Delta\alpha \|_\infty$	$\| \Delta h \|_2$	$\| \Delta h \|_\infty$
0.5	0.1	10	3.86×10^{-2}	8.22×10^{-3}	2.92×10^{-2}	4.89×10^{-3}
0.5	0.01	10	1.18×10^{-2}	7.69×10^{-4}	9.61×10^{-3}	5.10×10^{-4}
0.5	0.001	10	3.71×10^{-3}	7.64×10^{-5}	3.05×10^{-3}	5.12×10^{-5}
0.5	0.1	100	9.63×10^{-2}	8.22×10^{-3}	7.99×10^{-2}	5.14×10^{-3}

Table 8.1 Structural Model Validation Error Norms

to produce time histories for $0 \leq t \leq 10$. Figures 8.2a and b display the exact and numerical approximations of $\alpha(t)$ and $h(t)$ for the chosen time steps. Table 8.1 displays the maximum error norm and an L_2 error of the computed solutions. The maximum error observed in the integration of (8.10) over $0 \leq t \leq 10$, using $\Delta t = 0.01$, is 7.69×10^{-4}. As in the case of the linear structural model, this error is much smaller than errors sustained in the discretization and integration of the fluid-dynamic equations and the value of Δt is much larger than aerodynamic stability requires.

Table 8.2 displays the maximum error for a variation in f_s. The case $f_s/\alpha_s = 1$ describes a linear structural model and corresponds to the validation of Section 5.2.1. Only a slight variation in the maximum norm is observed for the full range of structural models and all errors are consistent with an $O(\Delta t^4)$ method.

f_s/α_s	$\| \Delta\alpha \|_\infty$	$\| \Delta h \|_\infty$
2.00	7.893×10^{-3}	4.807×10^{-3}
1.75	7.947×10^{-3}	4.813×10^{-3}
1.50	8.002×10^{-3}	4.821×10^{-3}
1.25	8.056×10^{-3}	4.832×10^{-3}
1.00	8.111×10^{-3}	4.845×10^{-3}
0.75	8.166×10^{-3}	4.860×10^{-3}
0.50	8.221×10^{-3}	4.890×10^{-3}
0.25	8.276×10^{-3}	4.919×10^{-3}
0.00	8.330×10^{-3}	4.952×10^{-3}

Table 8.2 Error Norms for a Variation in Bilinear Ratio: $\Delta t = 0.1$

Case#	α_s	f_s/α_s	Type	Method
66	0.5^o	0.5	Point	BGSN4
67	0.5^o	0.5-2.0	Boundary	BGSN4
68	0.25^o	0.5-2.0	Boundary	BGSN4
69	0.75^o	0.5-2.0	Boundary	BGSN4
70	1.0^o	0.5-2.0	Boundary	BGSN4

Table 8.3 Run Summary: Nonlinear Structural Hopf-Point Results

8.3 Nonlinear Structural Model Hopf-Point Results

The nonlinear structural model is applied to the NACA 64A006 PAPA model of the previous chapter. The following choice of structural and aerodynamic parameters are chosen to allow comparison with solutions in Chapter VII with a linear structural model and are considered the baseline:

$$x_{cg} = 0.375,\ x_\alpha = -0.25,\ \zeta_h = \zeta_\alpha = 0.1, \tag{8.19}$$

$$\frac{\omega_h}{\omega_\alpha} = 0.2,\ r_\alpha^2 = 0.25,\ \mu_s = 125,\ M_\infty = 0.85,\ \alpha_0 = 0^o, \tag{8.20}$$

and the G646-3 grid is used for all runs. Solutions are computed for wide ranges of α_s and f_s. Table 8.3 summarizes the cases for this section. The Hopf-point is obtained using a solution from the linear structural model of the previous chapter as an initial guess.

Using the BGSN4 Hopf-point and time-integration algorithms, Hopf curves are computed for $f_s/\alpha_s = 0.5$, 1.0 and 1.5. Figure 8.3 depicts the Hopf curve for the nonlinear and linear solutions.

Time-integration solutions follow the procedure discussed in Chapter VII; a converged static airfoil solution at a non-equilibrium angle-of-attack is computed and used as the initial condition. The airfoil is not in equilibrium causing it to oscillate. Solutions are stopped when α_{LC} changes less than 1×10^{-4} radians.

As one can see in Figure 8.3, for f_s/α_s decreased from 1.0 to 0.5, the Hopf-point changes by $\Delta\bar{u}^* = -2.71$. This change is significant when considering only a $\pm 0.5^o$ bilinear region, and can be described as highly destabilizing. The shape of the Hopf curve also changes. The Hopf curve for $f_s/\alpha_s = 1.0$ has a concave appearance (typical), whereas the $\pm 0.5^o$ region of nonlinearity has reversed the curvature near $\alpha_{LC} = 0.5^o$, making a more complex shape. Reducing f_s can then be described as having two effects, a shift of the complete Hopf curve and a change in the concavity of the Hopf curve near α_s.

Increasing f_s/α_s to 1.5 has a stabilizing effect on the Hopf-point and also changes the shape of the curve. The critical reduced velocity is $\bar{u}^* = 9.139$, a +30% shift from the linear value. The curve appears to develop a cusp at $\bar{u} = \bar{u}^*$, reducing α_{LC} for a given increase in $\bar{u}$ over $\bar{u}^*$, as compared to the linear Hopf curve.

Figure 8.4 depicts the flutter boundary associated with $\alpha_s = 0.5^o$ and a range of f_s/α_s ($0.5 \leq f_s/\alpha_s \leq 1.5$). The flutter boundary is obtained using previously computed solutions as initial guesses for solution points with different values of f_s. In general, for f_s greater than the linear value, the under-relaxation parameter $\omega = 0.5$ can be used with the grid chosen for this study. For f_s less than the linear value, the under-relaxation parameter must be reduced for stability. Solutions for $f_s/\alpha_s < 0.5$ were not attempted, since $\omega < 0.01$ is necessary for these cases.

Figure 8.4 also displays flutter boundaries for values of α_s equal to 0.25^o, 0.75^o, and 1.0^o. Each of the curves are limited from below to $f_s = 0.5\alpha_s$ for stability. As seen by Figure 8.4, certain combinations of f_s and α_s produce the same $\bar{u}^*$. To determine if a normalization variable

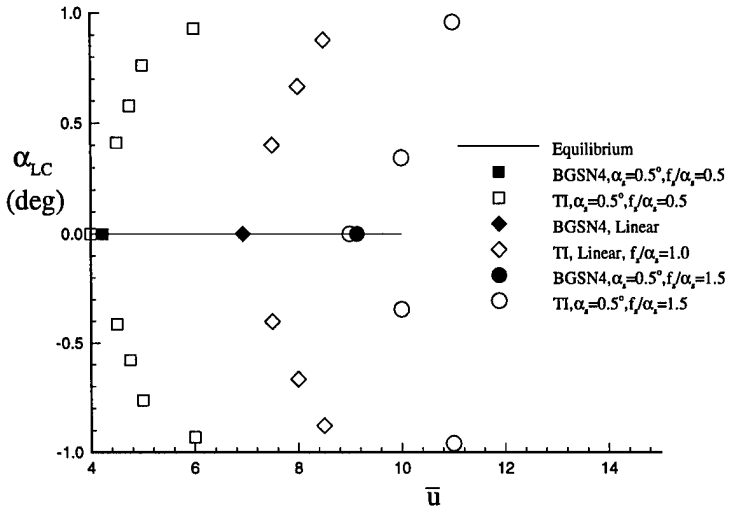


Figure 8.3 Hopf Curve: $\alpha_s = 0.5^o$, $M_\infty = 0.85$

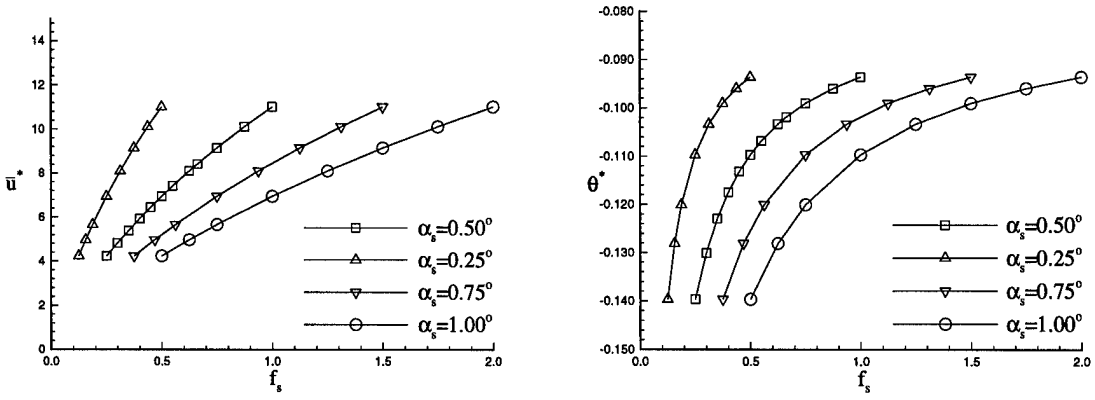


Figure 8.4 Flutter Boundaries: $M_\infty = 0.85$

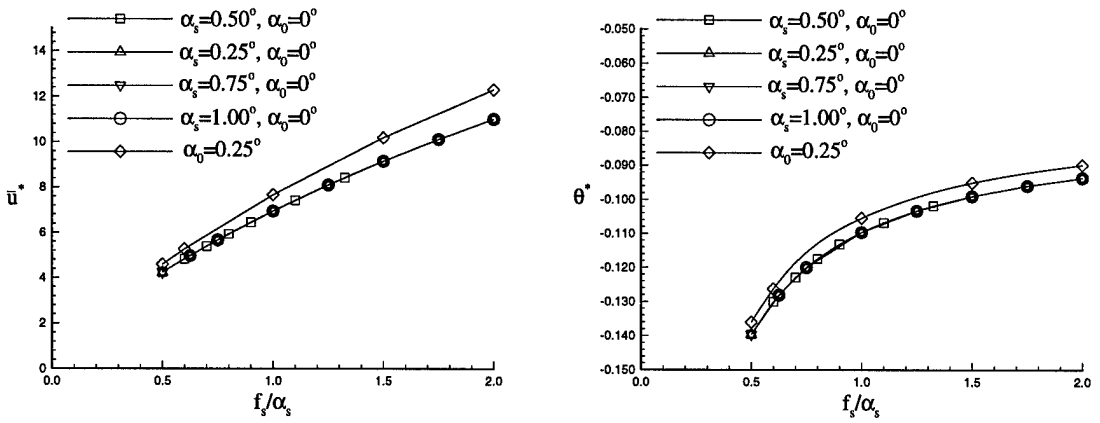


Figure 8.5 Flutter Boundary: $f_s/\alpha_s = 0.5 - 2.0$, $M_\infty = 0.85$

exists for the system, f_s/α_s is plotted versus $\bar{u}^*$ in Figure 8.5. All of the curves collapse onto a single curve, verifying f_s/α_s is a normalization variable of the system with a nonlinear structural model. This normalization variable is further verified by analyzing (8.6) and (8.7). At equilibrium, $f(\alpha) = \frac{f_s}{\alpha_s}(\alpha - \alpha_0)$ implying that f_s/α_s is more critical than either parameter independently.

Additionally, Hopf-points are computed for a nonzero value of the static pretwist, $\alpha_0 = 0.25^o$. This variation in α_0 allows assessment of changes to the normalization curve as α_0 is changed and demonstrates the applicability of BGSN5 to PAPA models with structural nonlinearities. Increasing the static pretwist to 0.25^o increases $\bar{u}^*$ and Θ^* for all values of f_s/α_s. The amount of increase in $\bar{u}^*$ lessens as f_s/α_s decreases.

IX. Conclusions and Recommendations

This chapter summarizes the accomplishments of the current research. First, a summary of the important results and the associated conclusions are presented, and then the recommendations for future research are discussed.

9.1 Summary and Conclusions

In Chapter I a specific set of objectives is provided that defines the scope of the research. This section provides conclusions from each area of the research and lists the objectives considered accomplished by the research.

9.1.1 Objective 1. In support of the first objective, a computer code capable of computing steady-state solutions with time-integration and also equilibrium solutions for a static airfoil at angle-of-attack and at transonic Mach number conditions is developed and validated. Validation of the code is accomplished by grid sensitivity analysis and through comparison with an AGARD solution [140].

Grid sensitivity is analyzed by varying four grid parameters: the domain size, R_{max}, the wall spacing, $\Delta wall$, the number of nodes around the airfoil, I, and the number of nodes normal to the airfoil, J. As R_{max} increases, the shock effectively moves downstream. Due to the type of boundary conditions chosen, the solution corresponding to the asymptotic value of C_l is achieved only after 150 chord lengths. Also, a 13.2% variation in C_l is observed from the smallest to largest domains computed. The wall spacing parameter produces a 4% increase in C_l for a change in $\Delta wall$ from 0.001 to 0.005. As the normal node distribution is refined, C_l monotonically increases, and the asymptotic value is effectively reached for $J = 32$. As the number of nodes around the airfoil is increased from 80 to 160, only a 2.8% increase in C_l is observed, implying that a grid of approximately $I = 100$ points adequately captures shocks with the present TVD scheme. R_{max} is

deemed the most limiting parameter since the asymptotic value of C_l is found for the domain sizes larger than 150 chord lengths (a 13.2% variation in C_l is observed).

All improvements in grid parameters increased C_l towards the AGARD solution. The AGARD solution is a 320 x 64 point "O"-grid with a 25 chordlength domain. At best, the current method compares within 6% of the AGARD solution. Good qualitative agreement and reasonable quantitative agreement with the AGARD solution gives confidence in the solution method and chosen grids.

The Newton-TVD method is used to compute solutions for both the NACA 0012 and NACA 64A006 airfoils at various Mach numbers and angles-of-attack. All solutions display crisp shocks where present and smooth Mach contours, implying that the TVD shock capturing method and chosen grid combination is capable of resolving the flowfield physics.

Lift, drag, and moment curves computed for the NACA 64A006 are typical of transonic airfoils and allow comparison for future research. Also, the efficiency with which the lift, drag, and moment curves are computed with manual continuation show the benefits of equilibrium methods.

Convergence properties of the Newton-TVD and TVD-time-integration solutions are documented and discussed. The time-integration and equilibrium methods are used to compute a solution to a residual norm of 10^{-5}. The time-integration method requires 50,000 iterations, whereas the equilibrium method requires only 11 iterations. After 10 Newton iterations and 20,000 explicit iterations, the solution is no longer changing, as measured by the peak Mach number. The convergence of Newton's method with approximate Jacobians is linear with a number of iterations commensurate with a quadratic method.

Jacobian matrices obtained with finite-difference approximations (called numerical Jacobians) are used to satisfy Objective 1's requirement of being easily extensible to viscous flow and fluid-structure interaction systems. The numerical Jacobian parameter ϵ_{jac} is an important parameter in the convergence of Newton's method. The convergence history for $\epsilon_{jac} = 10^{-5}$ has the best slope

and reaches the minimum residual value of 10^{-14}. The ϵ_{jac} values of 10^{-3}, 10^{-4}, 10^{-7}, and 10^{-8} all produce solutions that asymptote to a residual greater than 10^{-6}. Numerical Jacobian elements are adversely effected by truncation error if the value of ϵ_{jac} is chosen too large, and round-off error if the value of ϵ_{jac} is too small. All ϵ_{jac} values produce solutions which asymptote to residual norms less than 10^{-5}, the value determined to produce a converged flowfield solution. Also, after choosing the optimal value of ϵ_{jac} (10^{-5}) no difficulties have been encountered in computing solutions for wide ranges of aerodynamic and grid parameters.

The Newton's method approach developed in this research is thus efficient for computing solutions to transonic flows and easily extensible to other governing equations, due to the numerical Jacobians. The validation, analysis of convergence properties, and computed solutions are sufficient to consider Objective 1 satisfied.

9.1.2 Objective 2. In support of Objective 2, the code is extended to allow movement of the airfoil and mesh in pitch and plunge. Forced motion in pitch and plunge is computed and compared with an independent method to validate the moving airfoil modifications. A Beam-Warming code, ENS3DAE, is used for comparison. Peak C_l values are 5% higher for TVDntiAE than for ENS3DAE and peak C_m values are 15% higher for TVDntiAE than for ENS3DAE [22]. The Euler implicit time-integration scheme, larger time step, and the artificial dissipation model of ENS3DAE both contribute to a more dissipative scheme than TVDntiAE. The more dissipative scheme is consistent with smaller amplitude of oscillations for the dynamic motion. The good agreement between the two methods is sufficient to consider Objective 2 and the first two points of Objective 4 satisfied.

9.1.3 Objective 3 and Objective 4. In support of Objective 3, a pitch and plunge fluid-structure interaction model is implemented and validated. Validation of the pitch and plunge airfoil (PAPA) model is accomplished in two ways: Runge-Kutta integration of the structural model for a known solution, and comparison of a dynamic PAPA solution with two independent methods.

The first validation tool is a test of the structural equations implemented in the computer code to ensure an exact solution can be computed. Also, a determination of whether time steps governed by the aerodynamic equations produce accurate integration of the structural equations. The Runge-Kutta integration shows convergence of the numerically computed solution to the exact solution as time step is refined. Also, integration errors of the structural equations associated with time steps typical of aerodynamic equation stability limits are several orders of magnitude smaller than errors typical of discretization and integration of the fluid dynamics equations.

The flutter onset point of the PAPA system is computed with the current research code, ENS3DAE [22], and Kousen and Bendiksen [73] to validate the current method. The critical reduced velocity, $\bar{u}^*$, of TVDntiAE is less than 1% lower than the ENS3DAE solution and 3% lower than the Kousen and Bendiksen solution. TVDntiAE is a less dissipative model than both of the other methods, which is consistent with the lower value of $\bar{u}^*$. When the oscillation is forced, the amplitude of oscillation for $\bar{u} > \bar{u}^*$ is greater for TVDntiAE than for the other two methods. The quality of agreement between the three methods in simulating LCO is greatly reduced from the agreement in the flutter onset point, since differences in the time-integration schemes become more important in simulating LCO as peak values grow.

Another important test of the PAPA algorithm is a test of the consistency between equilibrium and steady-state solutions computed with the equilibrium and time-integration methods. A steady-state solution of the PAPA time-integration method is computed and compared with the corresponding PAPA equilibrium solution. The solutions are indistinguishable from each other for $t > 190$. This agreement is necessary for the solution methods to identify the same Hopf-point.

Equilibrium solutions of the PAPA model are computed for variations in the static pretwist, α_0, and the Mach number to document their effects. The static pretwist is varied from 0.25^o to 1.25^o with a fixed Mach number of 0.8. For a particular α_0, the airfoil pitches down from this value to a reduced angle ($\alpha_{eq} < \alpha_0$) due to the nose down pitching moment. As $\bar{u}$ increases, the

amount of pitch down increases, because increasing $\bar{u}$ effectively weakens the torsional spring. For a particular value of $\bar{u}$, as α_0 increases, the magnitude of the pitch down also increases due to the nose down pitching moment.

The Mach number is varied from 0.63 to 1.2 with $\alpha_0 = 1.25^o$. For a particular freestream Mach number, as $\bar{u}$ increases, the airfoil pitches down due to the increasing nose down pitching moment. In the range $0.63 \leq M_\infty \leq 0.95$, as M_∞ increases and $\bar{u}$ is held fixed, the airfoil pitches down. For $M_\infty = 1.2$, the equilibrium α is less than the equilibrium angle-of-attack for $M_\infty = 0.95$ which is related to the transonic flutter dip phenomenon. Through validation, consistency tests, and computed equilibrium solutions, Objectives 1 through 4 are considered satisfied for a linear structural model.

9.1.4 Objective 5. In support of Objective 5, analysis of the GR algorithm is performed to determine the workload when applied to the PAPA fluid-structure interaction system. The GR algorithm involves LU decomposition of a matrix G_Y and its square, G_Y^2. For systems with bordered rows and columns, such as fluid-structure interaction systems, G_Y^2 is a full matrix and requires $O(N^3)$ operations to LU decompose. Both the memory associated with storing a full matrix and the computational workload of decomposing the matrix make this method untenable for 2D airfoil problems with reasonably fine meshes. Therefore, a modified Hopf method, which is not dependent on a full $N \times N$ matrix, is developed.

9.1.5 Objective 6. The resulting algorithms, BGSN4 and BGSN5, are validated with eigenvalue analysis, time-integration, and a grid sensitivity analysis. The eigenvalue analysis consists of computing the eigenvalues of G_Y for various values of $\bar{u}$, and determining the particular reduced velocity for which a pair of complex eigenvalues has a zero real part. The computed eigenvalues produce a stability transition point within 0.05% of the Hopf point computed with BGSN4. Time-integration is also used to validate computation of the Hopf-point by BGSN4. All solutions with $\bar{u} < \bar{u}^*$ are damped oscillatory to a steady-state. Solutions with $\bar{u} > \bar{u}^*$ are stable limit cycles,

verifying the existence of a supercritical Hopf-bifurcation. The Hopf-point computed directly is in excellent agreement with the Hopf-point location bracketed by the time-integration analysis.

Grid effects on the computed Hopf-point are documented for a limited range of grid parameters. Increasing the number of points around the airfoil, reducing the spacing near the wall, and increasing the number of points normal to the airfoil all decrease the critical reduced velocity. Increasing the domain size, on the other hand, increases the critical reduced velocity. All variations in grid parameters performed result in less than an 11% variation in $\bar{u}^*$ from the baseline grid. Validation of BGSN4 and BGSN5 through eigenvalue analysis, time-integration, and the grid sensitivity analysis contribute to the completion of Objective 6.

Since the BGSN4 and BGSN5 algorithms are new, it is important to describe the convergence properties. The BGSN4 algorithm is compared against Newton's method for a coarse grid to determine the degradation of the method from the "theoretical best." Also, the BGSN4 and BGSN5 algorithms are compared against each other to determine the penalty of solving the full five-equation system. Finally, the BGSN5 method is used to compute a Hopf-point solution with various ϵ_{jac} values to determine the effect on convergence and accuracy.

The Newton's method solution for the $(3N+2) \times (3N+2)$ system is compared to the BGSN4 algorithm in full matrix form with $\omega = 1$. Convergence is nearly identical, with BGSN4 taking one additional iteration to converge. The BGSN4 solution with $\omega = 0.5$ takes 39 iterations to converge due to the under-relaxation, and the $\omega = 0.1$ solution takes 252 iterations to converge. The comparison with Newton's method implies that whenever under-relaxation values of 1.0 can be realized the BGSN algorithms practically recover the full matrix convergence properties.

The BGSN4 and BGSN5 algorithms are compared to each other for the same solution with $\omega = 1.0$. The minimum attainable residual norm is four orders of magnitude larger for BGSN5 than BGSN4. This is due to the numerical evaluation of $(G_Y P)_Y$ and the associated value of the ϵ_{jac} parameter. The effect of ϵ_{jac} on convergence and accuracy of the solution is assessed by

varying ϵ_{jac} from 10^{-3} to 10^{-8}. The minimum attainable residual is largest for $\epsilon_{jac} = 10^{-8}$ and smallest for $\epsilon_{jac} = 10^{-3}$ due to truncation error in the numerical Jacobians. The solution is not converged until $\epsilon_{jac} \geq 10^{-6}$, since G_Y is an element of the nonlinear system of equations. For this reason, a compromise between accuracy and convergence must be accomplished, or independent ϵ_{jac} parameters for G_Y and $(G_Y P)_Y$ must be used.

The BGSN4 and BGSN5 algorithms are used to compute a Hopf-point for the baseline grid to determine the computational workload. The BGSN4 solution is within 30% of the workload required to compute a regular point for the same number of iterations. It should be noted that, depending on the under-relaxation parameter, Hopf-points typically take 5-10 times the number of iterations of a regular point to converge. The BGSN5 algorithm requires six to seven times the computational time of a regular point for the same number of iterations due to the computation of $(G_Y P_1)_Y$ and $(G_Y P_2)_Y$. The Jacobian matrices $(G_Y P_1)_Y$ and $(G_Y P_2)_Y$ are much more computationally expensive than G_Y. A more efficient computation of G can improve relative performance since it is used in the computation of G_Y, $(G_Y P_1)_Y$, and $(G_Y P_2)Y$. Objective 5 is satisfied by BGSN4, and with improvements to the calculation of G, BGSN5 can also satisfy this objective.

9.1.6 Objective 7. The BGSN4 and BGSN5 algorithms are used to compute flutter boundaries for variations in Mach number, pitch and plunge damping, and static pretwist. The flutter boundaries are computed with manual continuation, which reduces the number of iterations for a particular flutter point by a factor of five. This savings allows the complete Mach and damping flutter boundaries to be computed in approximately 24 hours on a DEC 150 Mhz workstation. The static pretwist, which is computed with the BGSN5 algorithm, takes four days of computation on a DEC 150 Mhz workstation. It is believed that the efficiency of computing flutter boundaries has been greatly improved over explicit time-integration methods. All flutter boundaries are verified with time-integration.

The Mach flutter boundary has a rapid rise near $M_\infty = 0.85$ making it difficult to obtain solutions for $M_\infty > 0.855$. This behavior displays the weakness in computing flutter boundaries with manual continuation. The damping flutter boundary shows practically linear behavior over the range $0.1 \leq \zeta_{\alpha,h} \leq 0.5$. As damping is increased, the under-relaxation parameter can be increased. The author speculates that this is due to the damping terms on the diagonal of the Jacobian matrix.

The static pretwist flutter boundary is computed with BGSN5, since $\alpha_0 \neq 0$. This flutter boundary demonstrates the ability of the algorithm to compute nontrivial equilibrium and Hopf-point solutions. The flowfield must respond to changes in reduced velocity as the solution converges as opposed to the symmetric cases where the aerodynamic solution is unchanged as $\bar{u}^*$ varies. The flutter boundaries computed and verified with time-integration imply successful completion of Objective 1 through Objective 7 using a linear structural model.

The code is modified to allow a class of nonlinear structural models. The modified PAPA model is validated by comparison with an exact solution of the structural equations. Various time steps are used to determine if the solution converges to the exact solution as time step is improved and to determine if the nonlinearity in the structural model adds a restriction to the time step greater than the aerodynamic stability restriction. For a constant (α_s, f_s) pair, the error associated with integration of the structural equations is much smaller than errors in spatial discretization and integration of the aerodynamic equations and consistent with an $O(\Delta t^4)$ method. The flutter point of a nonlinear structural model with $\alpha_s = 0.5^o$ and $f_s = 0.25$ reduces the reduced velocity by 2.71 or 40% as compared to the linear structural model. This movement is significant for only a $\pm 0.5^o$ nonlinear region. The shape of the Hopf curve associated with $\alpha_s = 0.5^o$ and $f_s/\alpha_s = 0.5$ also changes near the value of α_s. The flutter point for $\alpha_s = 0.5^o$ and $f_s/\alpha_s = 1.5$ increases the critical reduced velocity by 32%. The shape of the associated Hopf curve can be described as having a cusp centered about $\alpha_{LC} = 0^o$.

A family of flutter boundaries are computed for variations in the structural parameters α_s and f_s. All solutions collapse to a single flutter boundary when plotted as f_s/α_s versus $\bar{u}^*$. This implies that the size of the nonlinear region is not important; only the slope of the torsional moment curve at equilibrium is significant. Therefore, a reduced number of solutions are necessary to determine the flutter boundary of a nonlinear structural model in this class. The flutter boundaries presented are sufficient to determine that all seven objectives are satisfied for BGSN4 and with relatively minor changes it is believed that the BGSN5 algorithm will satisfy all objectives as well.

9.2 Recommendations for Further Research

Three major areas of improvement to this work are thought to be necessary: aerodynamic modeling improvements, structural modeling improvements, and algorithm improvements. The major improvement in the aerodynamic model should be the addition of viscous terms and a variety of turbulence models. Although the aerodynamic model incorporates the Euler equations for the current study, the method is not limited to this equation set. Addition of viscous terms is straightforward due to the numerical calculation of Jacobian elements. Implementing the Navier-Stokes equations in the PAPA model is recommended for future research to document the change in the flutter boundary with shock-wave boundary-layer interaction. The viscous model should incorporate several turbulence models to assess their impact on the flutter boundaries. Another recommended improvement to the aerodynamic model is to implement more sophisticated farfield boundary conditions. Nonreflecting characteristic farfield boundary conditions help in alleviating the need for very large domains, shown to be the most sensitive grid parameter of the current method.

An area of interest not covered by the current research is deformation of the airfoil shape and the effects on the flutter boundary. This extension of the structural model allows more complex motion and is closer to the state of the art in the field of aeroelasticity. Another extension of the

structural model is to add a trailing-edge flap so that the phenomenon of trailing edge buzz can be analyzed with the modified Hopf algorithm.

Algorithm improvements are necessary if more complex aerodynamic and structural models are implemented which increase computational costs. Improvements in the computation of G will have a significant impact in the overall performance due to the numerical Jacobian computation. Also, implementing a more sophisticated continuation procedure can improve both the equilibrium solution computation and allow more complete flutter boundary calculations. Finally, improvements in the algorithm to allow solutions with no structural damping to be computed are important to improve performance and provide more complete flutter boundaries.

Appendix A. Harten-Yee Total Variation Diminishing Scheme

The Harten-Yee [149] total variation diminishing scheme (TVD) is presented in this appendix. First, the one-dimensional form is developed and applied to a one-dimensional model problem. Next, the scheme is extended to include two-dimensions with generalized coordinates. Also, modifications to the scheme to allow moving meshes is provided.

A.1 One-Dimensional TVD Scheme

The first-order accurate TVD scheme is developed by taking the hyperbolic conservation laws for the gas dynamic equations,

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0, \tag{A.1}$$

$$U = \begin{bmatrix} \rho \\ \rho u \\ E_t \end{bmatrix}, \quad F(U) = \begin{bmatrix} \rho u \\ (\rho u^2 + p) \\ (E_t + p)u \end{bmatrix}, \tag{A.2}$$

and rewriting with difference operators for constant temporal and spatial node spacing, $(\Delta t, \Delta x)$, to obtain

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} + \frac{F_{i+\frac{1}{2}}^n - F_{i-\frac{1}{2}}^n}{\Delta x} = 0. \tag{A.3}$$

Solving (A.3) for U_i^{n+1} one obtains

$$U_i^{n+1} = U_i^n - \lambda(F_{i+\frac{1}{2}}^n - F_{i-\frac{1}{2}}^n), \tag{A.4}$$

where $\lambda \equiv \frac{\Delta t}{\Delta x}$. TVD schemes modify the *flux function* F to enhance shock capturing, and to ensure physical solutions:

$$U_i^{n+1} = U_i^n - \lambda(\tilde{F}_{i+\frac{1}{2}}^n - \tilde{F}_{i-\frac{1}{2}}^n). \tag{A.5}$$

The *modified flux* is defined as follows:

$$\tilde{F}^n_{i+\frac{1}{2}} = \frac{1}{2}(F^n_i + F^n_{i+1}) + R^n_{i+\frac{1}{2}}\Phi^n_{i+\frac{1}{2}} \tag{A.6}$$

where $R_{i+\frac{1}{2}}$ is composed of rows of eigenvectors, a^l, of the Jacobian matrix F_U evaluated at $i+\frac{1}{2}$ and $\Phi = (\phi^1, \phi^2, \phi^3)^T$ with

$$\phi^l_{i+\frac{1}{2}} = \sigma(a^l_{i+\frac{1}{2}})\,(g^l_{i+1} + g^l_i) - \psi(a^l_{i+\frac{1}{2}} + \gamma^l_{i+\frac{1}{2}})\alpha^l_{i+\frac{1}{2}}. \tag{A.7}$$

The term $\alpha^l_{i+\frac{1}{2}} = {R^{-1}}^l_{i+\frac{1}{2}}\Delta_{i+\frac{1}{2}}U^l$ where $\Delta_{i+\frac{1}{2}}U^l = U^l_{i+1} - U^l_i$ and ${R^{-1}}^l_{i+\frac{1}{2}}$ are the left eigenvectors. The term ψ is the entropy function, which ensures that only physical solutions are computed, and is defined:

$$\psi(z) = \begin{cases} \frac{(z^2+\delta_1^2)}{2\delta_1} & \text{for } |z| \le \delta_1 \\ |z| & \text{for } |z| > \delta_1 \end{cases}. \tag{A.8}$$

The positive parameter δ_1 must be bounded by $0 \le \delta_1 \le 1$ for stability [148]. The term σ is defined by

$$\sigma(z) = \frac{1}{2}\left[\psi(z) - \lambda z^2\right] \tag{A.9}$$

with λ equal to $\Delta t/\Delta x$. The term g^l is a limiter function associated with the l^{th} eigenvalue. The particular limiter chosen is the minmod limiter:

$$g^l_i = \hat{S} \cdot max[0, min(\left|\alpha^l_{i-\frac{1}{2}}\right|, \hat{S} \cdot \alpha^l_{i+\frac{1}{2}})], \tag{A.10}$$

$$\hat{S} = sign(\alpha^l_{i-\frac{1}{2}}). \tag{A.11}$$

The term $\gamma^l_{i+\frac{1}{2}}$ is called the mean characteristic speed induced by g^l [148] and is defined by

$$\gamma^l_{i+\frac{1}{2}} = \begin{cases} (g^l_{i+1,j} - g^l_{i,j})/\alpha^l_{i+\frac{1}{2},j} & \text{if } \alpha^l_{i+\frac{1}{2},j} \neq 0 \\ 0 & \text{if } \alpha^l_{i+\frac{1}{2},j} = 0 \end{cases} . \tag{A.12}$$

Equation (A.12) requires a conditional statement which can be inefficient on vector machines. Moran [93] developed an expression which approximates (A.12) to slightly above machine precision for ϵ_m slightly above machine precision;

$$\gamma^l_{i+\frac{1}{2}} = (g^l_{i+1} - g^l_i)\left(\frac{\alpha^l_{i+\frac{1}{2}}}{\alpha^{l\,2}_{i+\frac{1}{2}} + \epsilon_m}\right). \tag{A.13}$$

Information is explicitly known on the node i instead of between the nodes at $i \pm \frac{1}{2}$, which requires some form of averaging. Roe [108] uses a special form of averaging that has the computational advantage of perfectly resolving stationary discontinuities for the gas dynamics equations. *Roe averaging* takes the following form

$$\hat{u}_{i+\frac{1}{2}} = \frac{Du_{i+1} + u_i}{D+1} \tag{A.14}$$

$$D = \left(\frac{\rho_{i+1}}{\rho_i}\right)^{\frac{1}{2}} \tag{A.15}$$

and is used for each of the unknowns in R, R^{-1}, and α. The system of equations (A.5)-(A.15) is a second-order-accurate, upwind, shock capturing scheme for the one-dimensional hyperbolic conservation laws. Yee [148] proposes to use (A.5) to compute successive time levels until convergence is achieved for a steady-state computation.

A.2 One-dimensional TVD Model Problem

An objective of the research is to compute steady-state solutions for the airfoil at transonic speeds. Towards this end, a model problem with a steady-state solution and a shock wave was

chosen. This model problem is an axisymmetric nozzle with inviscid walls. In the following sections the application of the previously described scheme to the model problem is described. Results and conclusions are presented.

A.2.1 Governing Equations. The average of the flow in an axisymmetric nozzle with a cross sectional area, $A(x)$ satisfies the one-dimensional system of equations

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = S(U) \tag{A.16}$$

$$U = \begin{bmatrix} \rho A \\ mA \\ E_t A \end{bmatrix}, \quad F(U) = \begin{bmatrix} mA \\ (mu + p)A \\ (E_t + p)uA \end{bmatrix}, \quad S(U) = \begin{bmatrix} 0 \\ p\frac{dA}{dx} \\ 0 \end{bmatrix} \tag{A.17}$$

where ρ is the density; $m = \rho u$ is the x-momentum; and E_t is the total energy per unit mass. The equations in system (A.16) are the continuity, momentum, and energy equations in conservative form with the assumption of inviscid flow. The chosen geometry, $A(x)$, is depicted in Figure A.1 and is described by

$$A(x) = 1.398 + 0.347 \ \tanh(0.8x - 4). \tag{A.18}$$

By analyzing the characteristic field, one can determine which boundary conditions are appropriate to specify and which conditions must be determined from the internal flow. For supersonic inlet conditions, three characteristics come from outside the computational domain, indicating that three flow properties may be specified at the inlet: ρ_{in}, m_{in}, and $E_{t_{in}}$. At the outflow boundary, only one characteristic comes from outside (downstream); and two come from inside the computational domain. This allows for one of the the variables, ρ, m, or E_t to be specified while the other two conditions must either be extrapolated from the interior or determined from characteristic compatibility conditions. The density at the outlet, ρ_{out}, is specified for ease of implementation. Both

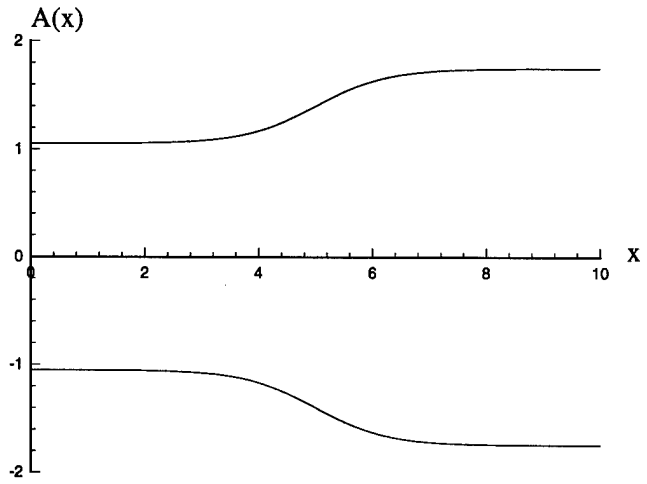


Figure A.1 Quasi-One Dimensional Nozzle Geometry

extrapolation and characteristic boundary conditions are used to determine their effect on accuracy and convergence. Extrapolation was implemented by enforcing a constant slope to second-order accuracy at the last interior point, resulting in

$$m_{out} = 2m_{out-1} - m_{out-2}, \tag{A.19}$$

$$E_{t_{out}} = 2\ E_{t_{out-1}} - E_{t_{out-2}}. \tag{A.20}$$

A.2.2 Application of Numerical Scheme. The system of equations (A.5) – (A.15) is a general scheme for hyperbolic conservation laws in one space dimension. This section will discuss tailoring the scheme for the model problem described in the previous section. Equation (A.5) was derived from the hyperbolic conservation laws without a source term. To include the source term, start with (A.16) and perform the same procedure as contained in (A.3)– (A.5) to obtain

$$U_i^{n+1} = U_i^n - \lambda \left(\tilde{F}_{i+\frac{1}{2}} - \tilde{F}_{i-\frac{1}{2}} + \Delta x S_i \right), \tag{A.21}$$

where S_i is defined in (A.17) and (A.6) –(A.15) are used as previously described. To evaluate (A.6), $a^l_{i+\frac{1}{2}}$, $R^l_{i+\frac{1}{2}}$, and ${R^{-1}}^l_{i+\frac{1}{2}}$ are needed. These terms are obtained by first obtaining the Jacobian matrix F_U:

$$F_U = A(x) \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{2}(\gamma-1)u^2 - u^2 & (3-\gamma)u & (\gamma-1) \\ u\left[\frac{1}{2}(\gamma-1)u^2 - H\right] & H-(\gamma-1)u^2 & \gamma u \end{bmatrix}. \tag{A.22}$$

The total enthalpy, H, is given by

$$H = (E_t + p)\frac{1}{\rho} = \frac{c^2}{\gamma-1} + \frac{1}{2}u^2. \tag{A.23}$$

The eigenvalues of F_U and the associated right eigenvectors are

$$a = \begin{bmatrix} u-c \\ u \\ u+c \end{bmatrix} \tag{A.24}$$

$$R = A(x) \begin{bmatrix} 1 & 1 & 1 \\ u-c & u & u+c \\ H-uc & \frac{1}{2}u^2 & H+uc \end{bmatrix}. \tag{A.25}$$

The l^{th} column of the R matrix relates to the l^{th} row in the eigenvalue vector a. To compute α, R^{-1} is needed:

$$R^{-1} = \frac{1}{A(x)} \begin{bmatrix} \frac{1}{2}(b_1 + \frac{u}{c}) & -\frac{1}{2}(ub_2 + \frac{1}{c}) & \frac{1}{2}b_2 \\ 1-b_1 & b_2 u & -b_2 \\ \frac{1}{2}(b_1 - \frac{u}{c}) & -\frac{1}{2}(ub_2 - \frac{1}{c}) & \frac{1}{2}b_2 \end{bmatrix}. \tag{A.26}$$

Knowing R^{-1}, α can be computed giving

$$\alpha = \begin{bmatrix} \frac{1}{2}(b_1 + \frac{u}{c})\Delta\rho - \frac{1}{2}(ub_2 + \frac{1}{c})\Delta m + \frac{1}{2}b_2\Delta E_t \\ (1 - b_1)\Delta\rho + b_2 u\Delta m - b_2\Delta E_t \\ \frac{1}{2}(b_1 - \frac{u}{c})\Delta\rho - \frac{1}{2}(ub_2 - \frac{1}{c})\Delta m + \frac{1}{2}b_2\Delta E_t \end{bmatrix}, \tag{A.27}$$

where

$$b_1 = (\gamma - 1)\frac{u^2}{2c^2}, \tag{A.28}$$

and

$$b_2 = (\gamma - 1)\frac{1}{c^2}. \tag{A.29}$$

The dependent variables of a, R, and α are Roe averaged quantities as previously described. Newton's method is used to solve

$$\tilde{F}_{i+\frac{1}{2}} - \tilde{F}_{i-\frac{1}{2}} + \Delta x S_i = 0, \tag{A.30}$$

the time-independent form of (A.21). Equation (A.30) represents four equations for each node value, i. In Section 2.3 the variable F in (2.18) is a vector of dimension $4n$ (where n is the number of nodes) and consists of a collocation of (A.30) for each i. A *function evaluation* refers to an evaluation of F. The explicit time-integration method, (A.21), following the method of Harten [50], is used to compute a steady-state solution and then compared to the Newton's method solution.

A.2.3 Results and Conclusions. Using the two codes developed previously, hereafter called QSHKNEW (Newton's method) and QSHKTIM (time-integration), several studies are made. An L_2 norm of the function evaluation is used to test for convergence, with a tolerance of 10^{-13}. The exact solution is computed with specified inlet conditions (ρ_{in}, m_{in}, E_{tin}) and an assumption of the location of the shock at $x = 5$. Isentropic relations are used to compute data prior to the shock

[1]. The inlet Mach number, M_{in}, for air (assumed to be a perfect gas) is calculated by

$$M_{in} = \frac{m_{in}}{\rho_{in}\left(1.4\ RT_{in}\right)^{\frac{1}{2}}}, \tag{A.31}$$

$$RT_{in} = \frac{p_{in}}{\rho_{in}}, \tag{A.32}$$

$$p_{in} = (\gamma - 1)\left(E_{tin} - \frac{1}{2}\frac{m_{in}^2}{\rho_{in}}\right). \tag{A.33}$$

Equation (A.18) is used to compute the area at x locations in the nozzle, such as the shock position area, A_1. The reference area, A_1^*, can be computed by

$$A_1^* = A_{in}\left[\frac{216}{125}M_{in}\left(1 + \frac{M_{in}^2}{5}\right)^{-3}\right]. \tag{A.34}$$

The Mach number just upstream of the shock, M_1, can be computed with (A.34) by substituting M_1 in for M_{in} and A_1 for A_{in}. The reference area A_1^* is constant for any position upstream of the shock. The nonlinear equation resulting, with unknown M_1, can be solved iteratively. The exact solution for any given x position prior to the shock can be computed by substituting the area, A_i, into (A.34) and iteratively solving for the Mach number, M_i. The density for a given Mach number can be computed by

$$\rho_i = \left[\frac{\left(1 + \frac{M_{in}^2}{5}\right)^{\frac{7}{2}}}{\left(1 + \frac{M_i^2}{5}\right)^{\frac{5}{2}}}\right]\rho_{in}. \tag{A.35}$$

Normal shock relations [1] are then used to compute quantities across the shock at $x = 5$. The density ratio and the Mach number across the shock are computed by

$$\frac{\rho_2}{\rho_1} = \frac{6M_1^2}{M_1^2 + 5}, \tag{A.36}$$

$$M_2 = \left(\frac{M_1^2 + 5}{7M_1^2 - 1}\right)^{\frac{1}{2}}. \tag{A.37}$$

Case	Grid	ϵ	c
1	20	0.01	1.0
2	50	0.01	1.0

Table A.1 Summary of Computer Runs for QSHKTIM and QSHKNEW

The same isentropic relations are then used to compute quantities following the shock. The outlet density is computed at $x = 10$ and is then used with the inlet conditions to define the numerical boundary conditions for both numerical methods:

$$U_{in} = \begin{bmatrix} 0.5020 \\ 0.6521 \\ 1.3758 \end{bmatrix}, \quad \rho_{out} = 0.7519 \, . \tag{A.38}$$

The two computer codes QSHKNEW and QSHKTIM use the same subroutine to compute the function evaluation, F, consisting of (A.30) for each node. QSHKTIM evaluates F and then computes a solution vector at the next time level by (A.21). QSHKNEW evaluates F, uses F to compute elements of the Jacobian matrix, and then computes a Newton iterate by (2.18). Both codes compute solutions on a uniform grid and either linearly interpolate initial data between the boundary conditions or use a restart file. Table A.1 is a summary of the cases computed by both codes. Due to the necessity of a good initial guess for QSHKNEW, QSHKTIM is used to compute a restart solution with a norm of 10^{-2}. Both codes then use the restart file for comparison purposes. Figures A.2 and A.3 show typical convergence histories for QSHKNEW and QSHKTIM, respectively. As one can see, QSHKNEW converges quadratically, whereas, QSHKTIM converges linearly with *scalloping*. Scalloping has been observed in convergence histories of explicit TVD schemes in [149] for computations of flow about a blunt body with reflected shock waves. Results of an error norm based on a comparison of exact and numerically computed data at a node, and the number of iterations to convergence are presented in Table A.2 for all cases run.

Case	QSHKNEW		QSHKTIM	
	Iterations	Error	Iterations	Error
1	10	2.37×10^{-2}	3657	2.39×10^{-2}
2	8	1.46×10^{-2}	11879	6.55×10^{-2}

Table A.2 Convergence and Accuracy Summary of Cases Run

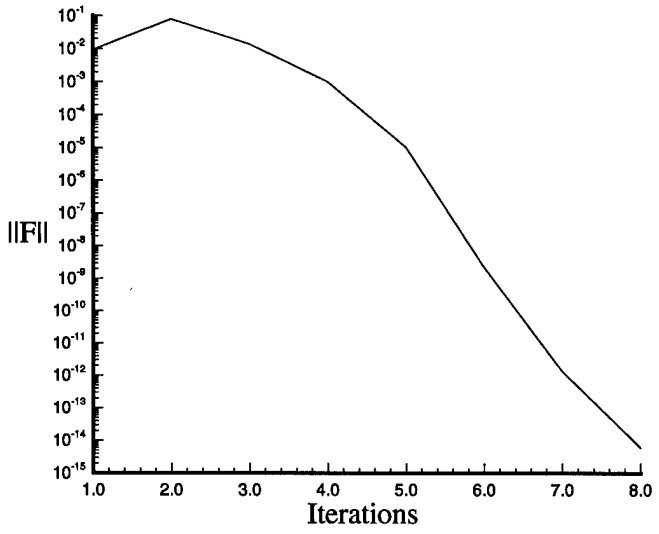


Figure A.2 Convergence History of QSHKNEW (Case 2)

In conclusion, it has been shown that Newton's method is a viable method of solving the nonlinear algebraic system of equations resulting from implementation of a TVD scheme for the Euler equations in one space dimension. Newton's method is in fact more robust than the explicit TVD time marching method. Newton's method is able to compute an accurate solution with extrapolation used for boundary conditions with no degradation of convergence rate or accuracy.

A.3 Two-Dimensional TVD Scheme

The Euler equations in two-dimensions are solved by employing an explicit algorithm that splits the multidimensional finite-difference algorithm into a sequence of one-dimensional opera-

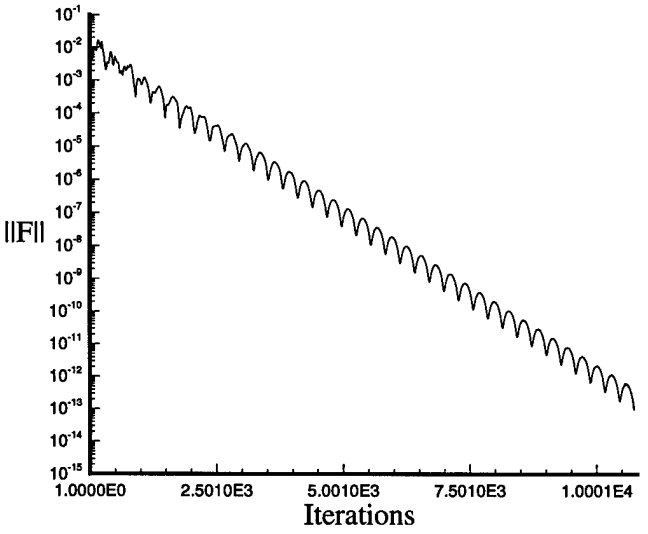


Figure A.3 Convergence History of QSHKTIM (Case 2)

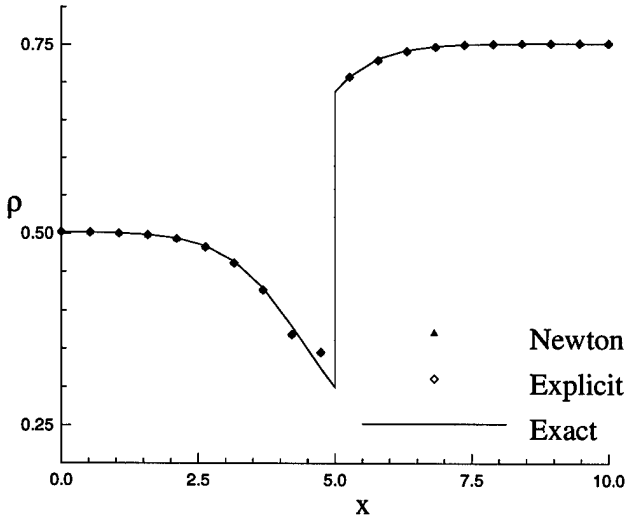


Figure A.4 Grid Sensitivity Study: 20 Nodes (Case 1)

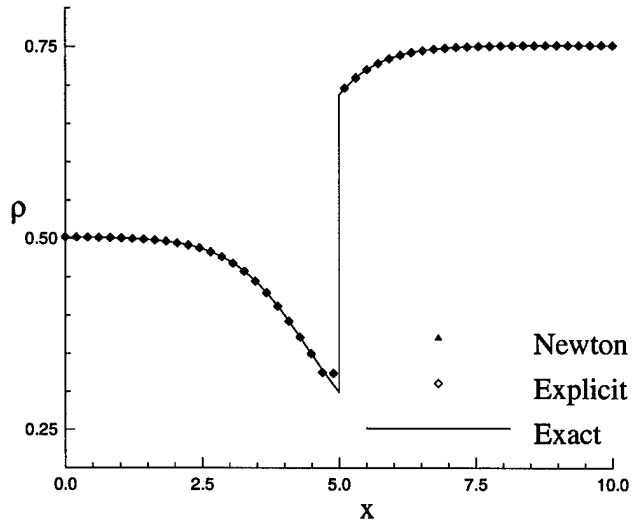


Figure A.5 Grid Sensitivity Study: 50 Nodes (Case 2)

tions [2]. The resulting algorithm is described as

$$L_\eta \hat{U}^n_{i,j} = \hat{U}^{n+1/2*}_{i,j} = \hat{U}^n_{i,j} - \Delta t \left(\tilde{\hat{G}}^n_{i,j+\frac{1}{2}} - \tilde{\hat{G}}^n_{i,j-\frac{1}{2}} \right), \tag{A.39}$$

$$L_\xi \hat{U}^{n+1/2}_{i,j} = \hat{U}^{n+1/2}_{i,j} - \Delta t \left(\tilde{\hat{F}}^{n+1/2}_{i+\frac{1}{2},j} - \tilde{\hat{F}}^{n+1/2}_{i-\frac{1}{2},j} \right), \tag{A.40}$$

with

$$\hat{U}^{n+1}_{i,j} = L_\xi L_\eta \hat{U}^n_{i,j}. \tag{A.41}$$

The TVD flux $\tilde{\hat{F}}_{i+\frac{1}{2},j}$ takes the form

$$\begin{aligned} \tilde{\hat{F}}_{i+\frac{1}{2},j} = \frac{1}{2} \Bigg[& (\xi_{x_{i,j}} \hat{F}_{i,j} + \xi_{y_{i,j}} \hat{G}_{i,j}) \frac{1}{J_{i,j}} \quad + \quad (\xi_{x_{i+1,j}} \hat{F}_{i+1,j} + \xi_{y_{i+1,j}} \hat{G}_{i+1,j}) \frac{1}{J_{i+1,j}} \\ & + \quad R_{\xi_{i+\frac{1}{2},j}} \Phi_{i+\frac{1}{2},j} \frac{1}{J_{i+\frac{1}{2},j}} \Bigg]. \end{aligned} \tag{A.42}$$

The TVD term $R_{\xi_{i+\frac{1}{2},j}}\Phi_{i+\frac{1}{2},j}$ is developed utilizing the local-characteristic approach, which is a generalization of Roe's approximate Riemann solver. The Jacobians $\hat{A}$ and $\hat{B}$ of $\hat{F}$ and $\hat{G}$ are required and can be written as

$$\hat{A} = (\xi_x A + \xi_y B)/J, \quad \hat{B} = (\eta_x A + \eta_y B)/J, \tag{A.43}$$

where $A \equiv F_U$, $B \equiv G_U$, and

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2}(\gamma-1)(u^2+v^2) - u^2 & (3-\gamma)u & (1-\gamma)v & (\gamma-1) \\ -uv & v & u & 0 \\ u\left[\frac{1}{2}(\gamma-1)(u^2+v^2) - H\right] & H-(\gamma-1)u^2 & (1-\gamma)uv & \gamma u \end{bmatrix}, \tag{A.44}$$

$$B = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -uv & v & u & 0 \\ \frac{1}{2}(\gamma-1)(u^2+v^2) - v^2 & (1-\gamma)u & (3-\gamma)v & (\gamma-1) \\ v\left[\frac{1}{2}(\gamma-1)(u^2+v^2) - H\right] & (1-\gamma)uv & H-(\gamma-1)v^2 & \gamma v \end{bmatrix}. \tag{A.45}$$

The total enthalpy, H, and the speed of sound, c, are given by

$$H = \frac{\gamma p}{(\gamma-1)\rho} + \frac{1}{2}(u^2+v^2), \quad c = \left(\frac{\gamma p}{\rho}\right)^{\frac{1}{2}}. \tag{A.46}$$

The eigenvalues of $\hat{A}$, denoted $a_\xi = (a_\xi^1, a_\xi^2, a_\xi^3, a_\xi^4)^T$, are

$$a_\xi = \begin{bmatrix} \xi_x u + \xi_y v - k_\xi c \\ \xi_x u + \xi_y v \\ \xi_x u + \xi_y v + k_\xi c \\ \xi_x u + \xi_y v \end{bmatrix} \tag{A.47}$$

where

$$k_\xi = \sqrt{{\xi_x}^2 + {\xi_y}^2}. \tag{A.48}$$

The right eigenvectors of $\hat{A}$, $R_\xi = (R^1_\xi, R^2_\xi, R^3_\xi, R^4_\xi)$, and the associated left eigenvectors, $R^{-1}_\xi = ({R^{-1}}^1_\xi, {R^{-1}}^2_\xi, {R^{-1}}^3_\xi, {R^{-1}}^4_\xi)$, are

$$R_\xi = \begin{bmatrix} 1 & 1 & 1 & 0 \\ u - k_1 c & u & u + k_1 c & -k_2 \\ v - k_2 c & v & v + k_2 c & k_1 \\ H - k_1 uc - k_2 vc & \frac{1}{2}(u^2 + v^2) & H + k_1 uc + k_2 vc & k_1 v - k_2 u \end{bmatrix}, \tag{A.49}$$

$$R^{-1}_\xi = \begin{bmatrix} (b_1 + k_1 u/c + k_2 v/c)/2 & (-b_2 u - k_1/c)/2 & (-b_2 v - k_2/c)/2 & b_2/2 \\ 1 - b_1 & b_2 u & b_2 v & -b_2 \\ (b_1 - k_1 u/c - k_2 v/c)/2 & (-b_2 u + k_1/c)/2 & (-b_2 v + k_2/c)/2 & b_2/2 \\ k_2 u - k_1 v & -k_2 & k_1 & 0 \end{bmatrix}, \tag{A.50}$$

where

$$k_1 = \frac{\xi_x}{\sqrt{{\xi_x}^2 + {\xi_y}^2}}, \quad k_2 = \frac{\xi_y}{\sqrt{{\xi_x}^2 + {\xi_y}^2}} \tag{A.51}$$

$$b_2 = (\gamma - 1)/c^2, \quad b_1 = b_2(u^2 + v^2)/2. \tag{A.52}$$

The term $\Phi_{i+\frac{1}{2},j}$ in (A.42) is defined as

$$\Phi_{i+\frac{1}{2},j} = \left(\phi^1_{i+\frac{1}{2},j}, \phi^2_{i+\frac{1}{2},j}, \phi^3_{i+\frac{1}{2},j}, \phi^4_{i+\frac{1}{2},j}\right)^T, \tag{A.53}$$

where

$$\phi^l_{i+\frac{1}{2},j} = \sigma(a^l_{i+\frac{1}{2},j})\left(g^l_{i+1,j} + g^l_{i,j}\right) - \psi(a^l_{i+\frac{1}{2},j} + \gamma^l_{i+\frac{1}{2},j})\alpha^l_{i+\frac{1}{2},j}. \tag{A.54}$$

The term $\alpha^l_{i+\frac{1}{2},j} = {R^{-1}}^l_{i+\frac{1}{2},j}\Delta_{i+\frac{1}{2},j}U^l$ where $\Delta_{i+\frac{1}{2},j}U^l = U^l_{i+1,j} - U^l_{i,j}$ and ${R^{-1}}^l_{i+\frac{1}{2},j}$ are the left eigenvectors. The term ψ is the entropy function, which ensures that only physical solutions are computed, and is defined:

$$\psi(z) = \begin{cases} \frac{(z^2+\delta_1^2)}{2\delta_1} & \text{for } |z| \leq \delta_1 \\ |z| & \text{for } |z| > \delta_1 \end{cases} . \tag{A.55}$$

The positive parameter δ_1 must be bounded by $0 \leq \delta_1 \leq 1$ for stability [148]. The term σ is defined by

$$\sigma(z) = \frac{1}{2}\left[\psi(z) - \lambda z^2\right] \tag{A.56}$$

with λ equal to either $\Delta t/\Delta\xi$ or $\Delta t/\Delta\eta$. The term g^l is a limiter function associated with the l^{th} eigenvalue. The particular limiter chosen is the minmod limiter:

$$g^l_{i,j} = \hat{S} \cdot max[0, min(\left|\alpha^l_{i-\frac{1}{2},j}\right|, \hat{S} \cdot \alpha^l_{i+\frac{1}{2},j})], \tag{A.57}$$

$$\hat{S} = sign(\alpha^l_{i-\frac{1}{2},j}). \tag{A.58}$$

The term $\gamma^l_{i+\frac{1}{2},j}$ is called the mean characteristic speed induced by g^l [148] and is defined by

$$\gamma^l_{i+\frac{1}{2},j} = \begin{cases} (g^l_{i+1,j} - g^l_{i,j})/\alpha^l_{i+\frac{1}{2},j} & \text{if } \alpha^l_{i+\frac{1}{2},j} \neq 0 \\ 0 & \text{if } \alpha^l_{i+\frac{1}{2},j} = 0 \end{cases} . \tag{A.59}$$

If g^l is set to zero the scheme degenerates to a first order upwind TVD scheme. The modified flux function $\tilde{\hat{G}}$ is developed in like manner.

The eigenvalues and eigenvectors of $\hat{B}$ are obtained by replacing ξ in (A.47)–(A.51) with η:

$$a_\eta = \begin{bmatrix} \eta_x u + \eta_y v - k_\eta c \\ \eta_x u + \eta_y v \\ \eta_x u + \eta_y v + k_\eta c \\ \eta_x u + \eta_y v \end{bmatrix}, \tag{A.60}$$

$$k_\eta = \sqrt{{\eta_x}^2 + {\eta_y}^2}. \tag{A.61}$$

There is no change to (A.49) and (A.50), the change of variable occurs in the definitions of (A.51):

$$k_1 = \frac{\eta_x}{\sqrt{{\eta_x}^2 + {\eta_y}^2}}, \quad k_2 = \frac{\eta_y}{\sqrt{{\eta_x}^2 + {\eta_y}^2}}, \tag{A.62}$$

and

$$\alpha_{\eta_{i,j+\frac{1}{2}}} = {R_\eta^{-1}}_{i,j+\frac{1}{2}} \Delta_{i,j+\frac{1}{2}} U. \tag{A.63}$$

It is important to remember that the method requires determining a, R, and R^{-1} at cell interfaces, $i \pm \frac{1}{2}$ or $j \pm \frac{1}{2}$, which necessitates calculating the metrics at these locations. The cell interface metrics and Jacobian's are average values of cell centered values:

$$\xi_{x_{i+\frac{1}{2},j}} = \frac{1}{2}(\xi_{x_{i+1,j}} + \xi_{x_{i,j}}), \quad \xi_{y_{i+\frac{1}{2},j}} = \frac{1}{2}(\xi_{y_{i+1,j}} + \xi_{y_{i,j}}), \tag{A.64}$$

and

$$\eta_{x_{i,j+\frac{1}{2}}} = \frac{1}{2}(\eta_{x_{i,j+1}} + \eta_{x_{i,j}}), \quad \eta_{y_{i,j+\frac{1}{2}}} = \frac{1}{2}(\eta_{y_{i,j+1}} + \eta_{y_{i,j}}), \tag{A.65}$$

and

$$J_{i+\frac{1}{2},j} = \frac{1}{2}(J_{i+1,j} + J_{i,j}), \quad J_{i,j+\frac{1}{2}} = \frac{1}{2}(J_{i,j+1} + J_{i,j}). \tag{A.66}$$

The matrices $A \equiv \frac{\partial \bar{F}}{\partial U}$ and $B \equiv \frac{\partial \bar{G}}{\partial U}$ differ from (A.44) and (A.45) because of the terms relating to a moving mesh and are defined

$$A = \begin{bmatrix} -u + (u - u_g) & 1 & 0 & 0 \\ \frac{1}{2}(\gamma - 1)(u^2 + v^2) - u^2 & (2 - \gamma)u + (u - u_g) & (1 - \gamma)v & (\gamma - 1) \\ -uv & v & (u - u_g) & 0 \\ u\left[\frac{1}{2}(\gamma - 1)(u^2 + v^2) - H\right] & H - (\gamma - 1)u^2 & (1 - \gamma)uv & (\gamma - 1)u + (u - u_g) \end{bmatrix}, \tag{A.67}$$

$$B = \begin{bmatrix} -v + (v - v_g) & 0 & 1 & 0 \\ -uv & (v - v_g) & u & 0 \\ \frac{1}{2}(\gamma - 1)(u^2 + v^2) - v^2 & (1 - \gamma)u & (2 - \gamma)v + (v - v_g) & (\gamma - 1) \\ v\left[\frac{1}{2}(\gamma - 1)(u^2 + v^2) - H\right] & (1 - \gamma)uv & H - (\gamma - 1)v^2 & (\gamma - 1)v + (v - v_g) \end{bmatrix}. \tag{A.68}$$

The eigenvalues of $\hat{A}$, denoted $a_\xi = (a_\xi^1, a_\xi^2, a_\xi^3, a_\xi^4)^T$, are

$$a_\xi = \begin{bmatrix} \xi_x(u - u_g) + \xi_y(v - v_g) - k_\xi c \\ \xi_x(u - u_g) + \xi_y(v - v_g) \\ \xi_x(u - u_g) + \xi_y(v - v_g) + k_\xi c \\ \xi_x(u - u_g) + \xi_y(v - v_g) \end{bmatrix}. \tag{A.69}$$

The eigenvalues of $\hat{B}$ can be obtained by replacing ξ with η in (A.69). The definitions (A.48)-(A.52) are unchanged for the moving mesh formulation.

Appendix B. Numerical Jacobian

The computation of Jacobian elements with finite-difference approximations allow Newton's method to be applied to complex nonlinear discrete systems. Orkwis [98] successfully applied *Numerical Jacobians* to a Roe flux-difference splitting scheme. He found that in all cases the numerical Jacobians performed as well as analytically computed Jacobians.

This appendix presents the numerical Jacobian method used in the current research and applies it to a two-dimensional model problem. Also, more complex Jacobian matrices necessary for a Hopf-bifurcation point computation are presented.

B.1 Regular Point Jacobian Matrices

The complexity of TVD schemes make the analytical calculation of Jacobian elements untenable. Since even analytic Jacobians are only as accurate as the precision of the computing machine, the author proposes a numerical evaluation of Jacobian elements via a first-order-accurate, forward-difference:

$$\frac{\partial G_i}{\partial Y_j} = \frac{G_i(Y + \epsilon \hat{e}_j) - G_i(Y - \epsilon \hat{e}_j)}{\epsilon} + O(\epsilon^2) \tag{B.1}$$

The value of ϵ is some small number, generally on the order of the square root of machine precision and becomes the order of error of the Jacobian element [142]. The term $\hat{e}_j$ is the j^{th} standard basis element.

Finite-difference methods produce Jacobian matrices which are banded and sparse. To ensure computational efficiency, the elements of the Jacobian matrix which are known to be zero are not calculated. This can be done since finite-difference approximations to the Navier-Stokes equations have a *computational stencil* of variables which influence the calculation of the equations at the node. A computational stencil is the set of nodes which affect the calculation of an equation at a particular node. Any variables outside of the stencil have no effect on the equations at the node. This implies that a row of Jacobian elements (constant i, varying j in (B.1)) can be calculated by

Streamfunction-Vorticity Driven Cavity			
Grid	Analytic (CPU sec)	Numerical (CPU sec)	% Difference
25 x 25	37.0	40.9	10.7 %
50 x 50	433.3	479.4	10.6 %
85 x 85	3505.66	3553.88	1.37 %

Table B.1 Comparison of Numerical versus Analytic Jacobian Matrices

perturbing each variable in the stencil and evaluating (B.1), while all other elements in the row are zero. Calculating rows of the Jacobian matrix as previously described is a valid approach but it requires a separate computation of all associated equations, termed a *function evaluation* call, for each perturbed equation. Calculating columns, on the other hand, allows one function call for each column of the Jacobian, making a more efficient method.

It was necessary to determine if this numerical Jacobian method was efficient and did not degrade the quadratic convergence of Newton's method. To do this a two-dimensional model problem was used. The model problem consisted of the streamfunction-vorticity formulation of the driven cavity. Central differences with no artificial dissipation was used for the discretization of the governing equations. TVD discretization was not employed because of the difficulty in obtaining the analytical Jacobian elements. As one can see from Table B.1, the calculation of the numerical Jacobian adds a 10% inefficiency for coarse grids but only 1% for finer grids. The finest grid consisted of 14,450 degrees of freedom. An airfoil calculation may consist of 51,200 degrees of freedom for the grid depicted in Section 3.4, implying airfoil calculations should require approximately 1% of overhead from the numerical Jacobian calculation. The reduction in overhead, measured in percent of the total time, is due to the impact of Gaussian elimination as the number of nodes is increased for a two-dimensional problem. In any case, $< 10\%$ is very reasonable, considering the ease of coding and extreme difficulty, if not impossibility, of analytically computing the vast number of Jacobian elements for a TVD scheme.

B.2 Hopf-Point Jacobian Matrices

Several vectors and matrices are necessary for the Hopf bifurcation calculation which were not necessary for the regular point computation, such as $(G_Y P_1)_Y$. The Jacobian matrix $(G_Y P_1)_Y$ is computed to $O(\epsilon)$ by utilizing a Taylor series expansion twice giving the following relationship:

$$[(G_Y P_1)_Y]_{i,j} = \frac{G_i(Y + \epsilon \underline{P_1} + \epsilon \underline{e_j}) - G_i(Y + \epsilon e_j) - G_i(Y + \epsilon P_1 - \epsilon e_j) + G_i(Y - \epsilon e_j)}{2\epsilon^2} + O(\epsilon). \tag{B.2}$$

Appendix C. Bordering Algorithm

Occasionally, linear systems with mixed matrix structures appear in engineering problems. One such linear system involves a matrix with a large banded structure bordered by a small number of full rows and columns. It would be inefficient to solve this system with a full matrix linear system solver, since the majority of the matrix is banded. For this reason, a method of partitioning the banded terms from the bordered elements, allowing efficient solutions of the linear system with a banded matrix solver is presented. Methods similar to this have been used in other research problems [13].

The bordering algorithm is designed to solve linear systems with a special structure in the linear system. In Figure C.1 the A matrix is an $n \times n$ matrix which has a special structure (i.e. banded). The linear system resulting from the blocked matrices of Figure C.1 can be described as

$$Ax_1 + Bx_2 = b_1, \tag{C.1}$$

$$Cx_1 + Dx_2 = b_2. \tag{C.2}$$

The dimensions of A, B, C, and D are the edge values, either m or n, in Figure C.1. Premultiplying (C.1) by A^{-1} gives the relationship

$$x_1 + A^{-1}Bx_2 = A^{-1}b_1. \tag{C.3}$$

The columns of $A^{-1}B$ are computed by solving m linear systems of the form

$$A\alpha_i = B_i, \tag{C.4}$$

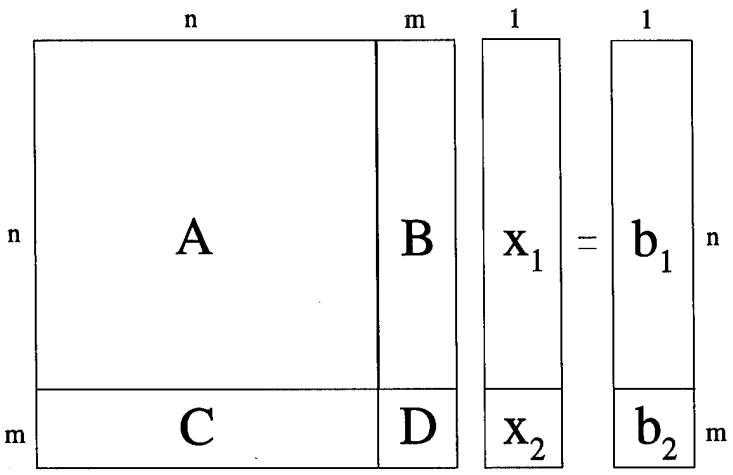


Figure C.1 Matrix Structure

where B_i is the i^{th} column of the matrix B and α_i is the i^{th} column of the matrix $A^{-1}B$. The vector $A^{-1}b_1$ is computed by solving the linear system

$$A\alpha_0 = b_1. \tag{C.5}$$

Equation (C.3) can then be rewritten as

$$x_1 + [\alpha_1, \cdots, \alpha_m]\, x_2 = \alpha_0. \tag{C.6}$$

Premultiplying (C.6) by C gives

$$Cx_1 + Kx_2 = C\alpha_0, \tag{C.7}$$

where

$$K = C\,[\alpha_1, \cdots, \alpha_m]\,. \tag{C.8}$$

Subtracting (C.7) from (C.2) gives the relationship

$$(D - K)x_2 = b_2 - C\alpha_0, \tag{C.9}$$

an $m \times m$ linear system for x_2. Knowing x_2, (C.6) can be used to obtain x_1. The following steps summarize the border algorithm:

- Solve $m + 1$ linear systems (C.4)-(C.5) with LU decomposition
- Compute the $m \times m$ matrix K using (C.8)
- Solve the $m \times m$ linear system (C.9) for x_2
- Compute x_1 by (C.6).

Appendix D. TVDntiAE Run Matrix and Grid Specification

This chapter provides a single source for all cases run in the document. All tables are extracted from the chapters without modification. Section D.1 provides the run tables and Section D.2 provides a summary of the computational time required to compute the solutions. Section D.3 provides a listing of the performance profiles for BGSN5 and BGSN4.

D.1 Run Matrix and Grid Specification Tables

See the following tables.

D.2 Computer Usage Estimate

Several computers were used to compute the solutions of the previous section. All are DEC Alpha machines with various CPU frequencies and bus structures. The following is a list of the machines used:

- DEC 4620/Alpha AXP 150 Mhz Dual Processor workstation with 384 Mb of RAM
- DEC 2100/Alpha AXP 190 Mhz Single Processor workstation with 128 Mb of RAM
- DEC Kubota/Cobra Alpha AXP 133 Mhz Single Processor workstation with 128 Mb of RAM
- DEC Alphastation 400 233 Mhz Single Processor workstation with 128 Mb of RAM
- DEC 3000/Alpha AXP 120 Mhz Single Processor workstation 64 Mb of RAM

The vast majority of the runs were completed on the first and second machines. Using performance estimates based on Chapter VII, 2500 hrs of computer resources is estimated.

Grid#	*I*	*J*	R_{max}	Δ_{wall}
NACA 0012 Airfoil				
G12-1	80	120	100	0.001
G12-2	80	80	50	0.001
G12-3	80	64	25	0.001
G12-4	80	60	20	0.001
G12-5	80	56	15	0.001
G12-6	80	42	10	0.001
G12-7	80	42	10	0.003
G12-8	80	42	10	0.005
G12-9	80	32	10	0.005
G12-10	80	16	10	0.005
G12-11	160	32	10	0.005
G12-12	120	32	10	0.005
G12-13	80	32	10	0.005
G12-14	80	32	10	0.005
G12-15	80	32	10	0.005
G12-16	80	32	10	0.005
G12-17	80	32	10	0.005
G12-18	320	64	25	0.001
NACA 64A006 Airfoil				
G646-1	100	31	15	0.015
G646-2	100	15	10	0.001
G646-3	100	15	10	0.015
G646-4	100	31	15	0.005
G646-5	100	15	10	0.005
G646-6	60	15	8	0.015
G646-7	60	15	8	0.005
G646-8	60	15	10	0.005
G646-9	60	13	4.1	0.005
G646-10	30	9	6	0.005

Table D.1 Grid Definitions (Table 3.1)

Case#	*Grid#*	I	J	R_{max}	Δ_{wall}	δ_1	ν_{cfl}	C_l	Type
1	G12-1	80	120	100	0.001	0	0.5	0.3432	EQ
2	G12-2	80	80	50	0.001	0	0.5	0.3286	EQ
3	G12-3	80	64	25	0.001	0	0.5	0.3211	EQ
4	G12-4	80	60	20	0.001	0	0.5	0.3173	EQ
5	G12-5	80	56	15	0.001	0	0.5	0.3084	EQ
6	G12-6	80	42	10	0.001	0	0.5	0.2997	EQ
7	G12-7	80	42	10	0.003	0	0.5	0.2935	EQ
8	G12-8	80	42	10	0.005	0	0.5	0.2877	EQ
9	G12-9	80	32	10	0.005	0	0.5	0.2871	EQ
10	G12-10	80	16	10	0.005	0	0.5	0.2700	EQ
11	G12-11	160	32	10	0.005	0	0.5	0.2954	EQ
12	G12-12	120	32	10	0.005	0	0.5	0.2918	EQ
13	G12-13	80	32	10	0.005	0.01	0.5	0.2871	EQ
14	G12-14	80	32	10	0.005	0.1	0.5	0.2871	EQ
15	G12-15	80	32	10	0.005	0.5	0.5	0.2871	EQ
16	G12-16	80	32	10	0.005	0	0.4	0.2871	EQ
17	G12-17	80	32	10	0.005	0	0.6	0.2871	EQ
18	G12-18	320	64	25	0.001	0	0.5	0.3432	TI

Table D.2 Case Definitions for Sensitivity Analysis (Table 4.1)

Case#	*Grid#*	M_∞	α	ϵ_{jac}	Type
19	G646-2	0.85	0^o	N/A	TI
20	G646-2	0.85	0^o	10^{-5}	EQ
21	G646-2	0.85	0^o	10^{-3}	EQ
22	G646-2	0.85	0^o	10^{-4}	EQ
23	G646-2	0.85	0^o	10^{-6}	EQ
24	G646-2	0.85	0^o	10^{-7}	EQ
25	G646-2	0.85	0^o	10^{-8}	EQ

Table D.3 Convergence Properties Run Summary (Table 4.3)

Case#	*Grid#*	x_{cg}	x_α	$\zeta_h = \zeta_\alpha$	r_α^2	ω_h/ω_α	μ_s	M_∞	α_0	$\bar{u}$	Type
38	G646-4	0.5	-0.2	0	0.29	0.2	10	0.87	0	1.9	TI
39	G646-4	0.5	-0.2	0	0.29	0.2	10	0.87	0	2.0	TI
40	G646-4	0.5	-0.2	0	0.29	0.2	10	0.87	0	2.1	TI
41	G646-4	0.5	-0.2	0	0.29	0.3434	10	0.87	0	2.25	TI
42	G646-5	0.375	-0.25	1	0.25	0.2	125	0.8	1.25^o	5.0	EQ
43	G646-5	0.375	-0.25	1	0.25	0.2	125	0.8	1.25^o	5.0	TI

Table D.4 Run Summary: PAPA Validation (Table 5.1)

Case#	*Grid#*	x_{cg}	x_α	$\zeta_h = \zeta_\alpha$	r_α	ω_h/ω_α	μ_s	M_∞	α_0	$\bar{u}$	Type
44	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	6.5	BGSN4
45	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	6.5	Eig
46	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	7.0	Eig
47	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	7.5	Eig
48	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	8.0	Eig
49	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	8.5	Eig
50	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	6.5	TI
51	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	7.0	TI
52	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	7.5	TI
53	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	8.0	TI
54	G646-7	0.375	-0.25	0.1	0.25	0.2	125	0.85	0^o	8.5	TI

Table D.5 Run Summary: Hopf-Point Validation (Table 7.1)

Case#	*Grid#*	I	J	$\Delta wall$	R_{max}	$\bar{u}^*$	Θ^*	% Diff $\bar{u}^*$	% Diff Θ^*
55	G646-9	60	13	0.005	4.1	6.434	-0.1194	−7.3%	8.8%
56	G646-7	60	15	0.005	8.0	7.050	-0.1079	1.6%	-1.6%
57	G646-8	60	15	0.005	10.0	7.588	-0.1048	9.4%	-4.5%
58	G646-6	60	15	0.015	8.0	7.720	-0.1055	11.3%	-3.8%
59	G646-5	100	15	0.005	10.0	6.674	-0.1123	−3.8%	2.4%
60	G646-3	100	15	0.015	10.0	6.937	-0.1097	Baseline	Baseline

Table D.6 Run Summary: Grid Refinement (Table 7.4)

Case#	*Grid#*	$\zeta_h = \zeta_\alpha$	M_∞	α_0	ω	Type	Method
61	G646-3	0.5	0.85	0	0.1	Point	BGSN4
62	G646-3	0.5	0.7-0.852	0	0.5	Boundary	BGSN4
63	G646-3	0	0.7-0.854	0	0.5	Boundary	BGSN4
64	G646-3	0-0.5	0.85	0	0.5	Boundary	BGSN4
65	G646-3	0.5	0.85	0^o-1.25^o	0.5	Boundary	BGSN4

Table D.7 Run Summary: Hopf-Point Results (Table 7.5)

Case#	α_s	f_s/α_s	Type	Method
66	0.5^o	0.5	Point	BGSN4
67	0.5^o	0.5-2.0	Boundary	BGSN4
68	0.25^o	0.5-2.0	Boundary	BGSN4
69	0.75^o	0.5-2.0	Boundary	BGSN4
70	1.0^o	0.5-2.0	Boundary	BGSN4

Table D.8 Run Summary: Nonlinear Structural Hopf-Point Results (Table 8.3)

D.3 BGSN5 and BGSN4 Profile Listing

This section presents the FORTRAN profile of the BGSN4 and BGSN5 Hopf-point codes. The code was profiled for the same grid and the same case. The solution was stopped after two iterations.

%time	seconds	cum %	cum sec	procedure (file)
19.6	62.9658	19.6	62.97	etasweep (TVDntiAE.f)
15.1	48.4482	34.8	111.41	matmuls (paclib46.f)
12.9	41.4951	47.7	152.91	jac (TVDntiAE.f)
9.2	29.5117	56.9	182.42	xisweep (TVDntiAE.f)
8.9	28.4824	65.8	210.90	blklum (paclib46.f)
8.0	25.6348	73.8	236.54	dtime (TVDntiAE.f)
5.4	17.1523	79.2	253.69	cut (TVDntiAE.f)
5.0	15.9922	84.2	269.68	energy (TVDntiAE.f)
4.0	12.9219	88.2	282.60	brdlum (paclib46.f)
3.4	10.9531	91.6	293.56	fcn (TVDntiAE.f)
1.8	5.6885	93.4	299.25	blkdec (paclib46.f)
1.7	5.3457	95.0	304.59	boundary (TVDntiAE.f)
1.2	3.9844	96.3	308.58	brdsol (paclib46.f)
1.0	3.3516	97.3	311.93	primitive (TVDntiAE.f)
0.5	1.7061	97.9	313.63	bordjac (TVDntiAE.f)
0.4	1.2217	98.3	314.86	derivs (TVDntiAE.f)
0.4	1.1729	98.6	316.03	gaussj (paclib46.f)
0.3	1.0254	98.9	317.05	modify (TVDntiAE.f)
0.2	0.7764	99.2	317.83	prdavbrd (paclib46.f)
0.2	0.5322	99.3	318.36	matinv (paclib46.f)
0.2	0.4932	99.5	318.86	metrics (TVDntiAE.f)
0.1	0.4355	99.6	319.29	gsbrd (paclib46.f)
0.1	0.2627	99.7	319.55	farfield (TVDntiAE.f)
0.1	0.2393	99.8	319.79	unblok (TVDntiAE.f)
0.1	0.2158	99.9	320.01	blok (TVDntiAE.f)
0.1	0.2090	99.9	320.22	border (paclib46.f)
0.0	0.1260	100.0	320.34	newgrd (TVDntiAE.f)
0.0	0.0742	100.0	320.42	gyppar (TVDntiAE.f)
0.0	0.0166	100.0	320.43	tvdntiae-main (TVDntiAE.f)
0.0	0.0098	100.0	320.44	mod2 (TVDntiAE.f)
0.0	0.0059	100.0	320.45	output (TVDntiAE.f)
0.0	0.0059	100.0	320.46	gpar (TVDntiAE.f)
0.0	0.0059	100.0	320.46	update (TVDntiAE.f)

Table D.9 Profile listing for BGSN4: generated Thu Apr 25 05:06:26 1996, each sample covers 4.00 byte(s) for 0.0003% of 320.4619 seconds

%time	seconds	cum %	cum sec	procedure (file)
26.8	383.8975	26.8	383.90	metrics (TVDntiAE.f)
14.2	203.9424	41.1	587.84	mod2 (TVDntiAE.f)
11.2	159.7676	52.2	747.61	etasweep (TVDntiAE.f)
10.6	152.0117	62.8	899.62	gypy (TVDntiAE.f)
5.5	78.2090	68.3	977.83	xisweep (TVDntiAE.f)
4.4	62.4355	72.7	1040.26	dtime (TVDntiAE.f)
4.0	57.7803	76.7	1098.04	cut (TVDntiAE.f)
3.6	51.8789	80.3	1149.92	newgrd (TVDntiAE.f)
3.4	48.4707	83.7	1198.39	matmuls (paclib46.f)
3.2	45.9326	86.9	1244.33	energy (TVDntiAE.f)
3.0	42.7959	89.9	1287.12	jac (TVDntiAE.f)
2.8	40.6494	92.8	1327.77	fcn (TVDntiAE.f)
2.3	32.2451	95.0	1360.02	boundary (TVDntiAE.f)
2.0	28.5488	97.0	1388.57	blklum (paclib46.f)
0.9	12.7998	97.9	1401.37	brdlum (paclib46.f)
0.5	7.5938	98.4	1408.96	primitive (TVDntiAE.f)
0.4	5.9951	98.8	1414.95	blkdec (paclib46.f)
0.3	3.9063	99.1	1418.86	brdsol (paclib46.f)
0.2	3.5205	99.4	1422.38	derivs (TVDntiAE.f)
0.1	1.7998	99.5	1424.18	modify (TVDntiAE.f)
0.1	1.7842	99.6	1425.96	prdavbrd (paclib46.f)
0.1	1.7100	99.7	1427.67	bordjac (TVDntiAE.f)
0.1	1.1660	99.8	1428.84	gaussj (paclib46.f)
0.1	0.7969	99.9	1429.64	farfield (TVDntiAE.f)
0.0	0.5195	99.9	1430.16	matinv (paclib46.f)
0.0	0.4980	99.9	1430.66	gsbrd (paclib46.f)
0.0	0.2529	100.0	1430.91	blok (TVDntiAE.f)
0.0	0.2500	100.0	1431.16	unblok (TVDntiAE.f)
0.0	0.2109	100.0	1431.37	border (paclib46.f)
0.0	0.0791	100.0	1431.45	gyppar (TVDntiAE.f)
0.0	0.0088	100.0	1431.46	tvdntiae (TVDntiAE.f)
0.0	0.0059	100.0	1431.46	output (TVDntiAE.f)
0.0	0.0059	100.0	1431.47	update (TVDntiAE.f)
0.0	0.0059	100.0	1431.47	gpar (TVDntiAE.f)

Table D.10 Profile listing for BGSN5: generated Thu Apr 25 06:33:38 1996, each sample covers 4.00 byte(s) for 0.0003% of 320.4619 seconds

Appendix E. TVDntiAE Code Summary

This appendix gives an overview of the computer code developed during the current research. First, a summary of the overall code is provided with a listing of the types of solutions obtainable. Next, a summary of the individual subroutines of TVDntiAE are provided. A description of a library of subroutines necessary to solve the linear systems is also provided. Finally, the archival structure of the code and the data generated is provided.

E.1 Code Overview

TVDntiAE is a FORTRAN 90 computer code developed to compute a suite of solutions for a two-dimensional airfoil with an "O"-grid. The aerodynamics model the code is based on is the Euler equations discretized with the Harten-Yee [149] total variation diminishing (TVD) scheme. The structural model is a two degree-of-freedom pitch and plunge model composed of linear and torsional springs.

The code is capable of computing the following types of solutions:

- Explicit time-integration to steady-state solution of a static airfoil,
- Newton's method fully implicit solution of a static airfoil,
- Time-accurate solution of an airfoil in forced pitch and plunge motion,
- Time-accurate solution of an airfoil with pitch and plunge structural coupling,
- Newton's method fully implicit equilibrium solution of an airfoil with pitch and plunge structural coupling,
- Hopf-bifurcation point solution of an airfoil with pitch and plunge structural coupling.

The fully implicit Jacobian matrix is computed numerically via central-difference approximations. Equilibrium and Hopf-point solutions utilize linear algebra subroutines of a library developed by

Beran [13], and extended to include the Hopf-bifurcation point algorithm by the author, called PACLIB (paclib46.f).

E.2 List of Subroutines

The following is a list of the subroutines in TVDntiAE.

- EXPLICIT-driving subroutine to compute a time-integration solution to steady-state.
- DYNAMIC-driving subroutine to compute a time-accurate time-integration solution for either a forced oscillation or PAPA model.
- NEWTON-driving subroutine to compute an equilibrium solution with PACLIB.
- FCN-subroutine to compute the nonlinear algebraic equation s for each node.
- XISWEEP-subroutine to sweep the xi direction with a TVD operator.
- ETASWEEP-subroutine to sweep the eta direction with a TVD operator.
- METRICS-subroutine to compute the metrics of the grid (only called once).
- BOUNDARY-subroutine to update the surface boundary values.
- FARFIELD-subroutine to compute the nonlinear algebraic equations at the farfield.
- CUT-subroutine to update the overlap points.
- DTIME-subroutine to compute the maximum time step for the stability criteria.
- GEOMETRY-subroutine to compute the slope angle of the surface at a grid point.
- PRIMITIVE-subroutine to compute the primitive variable from the conserved variables.
- UPDATE-subroutine to update the conserved variables from a Newton iterate.
- UPDATB-subroutine to update the structural variables from a Newton iterate.
- MODIFY-subroutine to modify a conserved variable by epsilon for the numerical Jacobian routine.

- MOD2-subroutine to modify a conserved variable by epsilon times the eigenvector for the numerical Jacobian routine
- OUTPUT-subroutine to output the solution vector.
- ENERGY-subroutine to compute the lift drag and moment coefficient and then compute the q vector for the structural model.
- FCN1-subroutine to compute the partial derivative of the aero equations with respect to par(1).
- FCN3-subroutine to compute the partial derivative of the aero equations with respect to par(3).
- FCN4-subroutine to compute the partial derivative of the aero equations with respect to par(4).
- JAC-subroutine to compute the numerical Jacobian of the function evaluation of FCN.
- BLOK-subroutine to block up an nktot vector into 16 element blocks.
- UNBLOK-subroutine to unblock a 16 element block vector into an nktot vector.
- FORCED-subroutine to drive a forced oscillation time-accurate.
- STRUC-subroutine to drive a PAPA time accurate solution.
- DERIVS-subroutine to compute the derivatives necessary for the structural equations.
- RK4-subroutine to perform fourth order Runge-Kutta taken from Numerical Recipes.
- RKD-subroutine to drive the fourth-order Runge-Kutta integration.
- BORDJAC-subroutine to compute the bordered Jacobian matrix.
- BCDCALC-subroutine to drive the calculation of B C D matrices needed in BORDJAC.
- CDCALC-subroutine to compute C and D Jacobian matrices needed in BORDJAC.
- BCALC-subroutine to compute the B matrix needed in BORDJAC.

- MATINV2-subroutine to compute the inverse of a square matrix.
- GUPUSUB-subroutine to compute the Jacobian matrix $[(FU)P]U$ needed in the Hopf point calculation.
- PACKAP-subroutine to pack a blocked banded Jacobian matrix into a fully packed banded Jacobian matrix.
- GYPY-subroutine to compute the bordered Jacobian matrix elements for the Hopf point calculation (calls GUPUSUB).
- GPAR-subroutine to compute the partial derivative of the aero and structural equations with respect to par(.).
- GYPPAR-subroutine to compute the partial derivative of the matrix GYP with respect to par(.).
- EIGOUT-subroutine to compute the set of eigenvalues of GY.
- BALANC-subroutine to balance the GY matrix (from Numerical Recipes).
- ELMHES-subroutine to put GY into upper Hessian form (from Numerical Recipes).
- HQR-subroutine to obtain the eigenvalues of a Hessian matrix (from Numerical Recipes).

E.3 PACLIB Overview

PACLIB is a library of FORTRAN subroutines to compute solutions of nonlinear systems of equations (in the current version FORTRAN 90 statements are used). The library is based on Newton's method with extensions to include: the chord method, pseudo-arclength continuation, and a modified form of the Griewank and Reddien algorithm of solving for Hopf-bifurcation points. All linear algebra problems of the form $Ax = b$ are solved assuming A is a sparse banded matrix in packed form and either utilizes Gaussian-elimination or LU decomposition. The library has been used to solve engineering problems in (at least) the following references: [13], [14], [15], [96], [94], [95],

and [135]. The subroutine GSBRD in PACLIB version 4.6 (paclib46.f) is the subroutine written by the author. It solves an extended system of equations for a Hopf-point with a blocked-Gauss-Seidel-Newton under-relaxation method.

E.4 Data Archival Structure

The complete set of data and computer codes and document files are archived on the machine called "data". The following is a description of each subdirectory containing pertinent files.

- */fluids/smorton/DISS* - dissertation **.tex*, **.sty*, **.fig*, and **.eps* files.
- */fluids/smorton/RENO* - 34^{th} AIAA Reno conference paper **.tex*, **.sty*, **.fig*, and **.eps* files.
- */fluids/smorton/SANDIEGO* - AIAA Applied Aero San Diego conference paper **.tex*, **.sty*, **.fig*, and **.eps* files.
- */fluids/smorton/DISS/DATA* - dissertation data files and associated tecplot v6.0 style, layout, and plot files
- */fluids/smorton/DISS/TVDNTIAE* - TVDntiAE paclib46 research codes developed for the dissertation along with equilibrium and time-accurate data post processing programs
- */fluids/smorton/DISS/TVDNTIAE/CHECKTI* - contains the input and output files of a time-integration run for checking TVDntiAE in this mode (including instructions on compilation and run commands)
- */fluids/smorton/DISS/TVDNTIAE/CHECKEQ* - contains the input and output files of an equilibrium run for checking TVDntiAE in this mode (including instructions on compilation and run commands)

- */fluids/smorton/DISS/TVDNTIAE/CHECKBGSN4* - contains the input and output files of a BGSN4 Hopf-point run for checking TVDntiAE in this mode (including instructions on compilation and run commands)
- */fluids/smorton/DISS/TVDNTIAE/CHECKBGSN5* - contains the input and output files of a BGSN5 Hopf-point run for checking TVDntiAE in this mode (including instructions on compilation and run commands)
- */fluids/smorton/DISS/TVDNTIAE/SOLUTIONS* - contains the solution file in binary format including conserved variable, grid, and structural data
- */fluids/smorton/DISS/TVDNTIAE/QSHK* - contains all files generated during quasi-one dimensional nozzle problem, including *.f and *.dat files

Each subdirectory includes a "readme" file explaining the contents of the directory.

Bibliography

1. Ames Research Staff. *Equations, Tables, and Charts for Compressible Flow.* Report 1135, National Advisory Committee for Aeronautics. Washington: AMTEC ENGINEERING INC. 1506-106th N. E. Bellevue, Washington.

2. Anderson, D. A.; Tannehill, J. C.; and Pletcher, R, H. *Computational Fluid Mechanics and Heat Transfer.* New York: Hemisphere Publishing Corporation, 1984.

3. Anderson, W. K.; Thomas, J. L.; and Rumsey, C. L.: "Application of Thin Layer Navier-Stokes Equations Near Maximum Lift." *AIAA 22nd Aerospace Sciences Meeting*, Reno Nevada, AIAA Paper 84-0049, January 1984.

4. Baldwin, B. S. and Lomax, H.: "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," *AIAA 16th Aerospace Sciences Meeting.* AIAA 78-257, 1978.

5. Ballahus, W. F. and Goorjian, P. M.: "Implicit Finite-Difference Computations of Unsteady Transonic Flows about Airfoils," *AIAA Journal, 15, No. 12*, (December 1977).

6. Barton, J. T. and T. H. Pulliam, T. H.: "Airfoil Computation at High Angles of Attack, Inviscid and Viscous Phenomena," *AIAA 22nd Aerospace Sciences Meeting.* AIAA 84-0524, 1984.

7. Barton, J. T. and Pulliam, T. H.: "Airfoil Computation at High Angles of Attack, Inviscid and Viscous Phenomena." *AIAA Journal, Vol. 24, No. 5*, May 1986.

8. Batchelor, G. *An Introduction to Fluid Dynamics.* New York: Cambridge University Press, 1967.

9. Belk, D. M., Janus, J. M., and Whitfield, D. L.: "Three Dimensional Unsteady Euler Equations Solutions on Dynamic Grids," AIAA paper No. 85-1704, July 1985.

10. Bendiksen, O. O. and Kousen, K.A.: "Transonic Flutter Analysis Using the Euler Equations," *AIAA Dynamics Specialists Conference, Monterey, CA, April 9-10, 1987."* AIAA-87-0911.

11. Bendiksen, O. O.: "Nonclassical Aileron Buzz in Transonic Flow," *34th Structures, Structural Dynamics and Materials Conference, La Jolla, CA, April 19-22, 1993."* AIAA-93-1479.

12. Bennett, R. M.; Eckstrom, C. V.; Rivera, J. A., Jr.; Dansberry, B. E.; Farmer, M. G.;Durham, M. H.: "The Benchmark Aeroelastic Models Program - Description and Highlights of Initial Results," NASA TM-104180, December 1991.

13. Beran, P. S. *An Investigation of the Bursting of Trailing Vortices Using Numerical Simulation.* PhD Dissertation. California Institute of Technology, Pasadena, CA. 1989.

14. Beran, P. S.: "Steady and Unsteady Solutions of the Navier-Stokes Equations for Flows About Airfoils at Low Speeds," *AIAA 22nd Fluid Dynamics, Plasma Dynamics, and Lasers Conference.* AIAA 91-1733, 1991.

15. Beran, P. S. and Lutton, M. J.: "Hopf Bifurcation in Viscous Flows About Airfoils at Low Speeds," *AIAA 22nd Fluid Dynamics, Plasma Dynamics, and Lasers Conference.* AIAA 91-1807, 1991.

16. Beran, P. S.: *Hopf-Bifurcation Lecture Notes*, Air Force Institute of Technology, Wright-Patterson AFB OH. Private Communication, 1994.

17. Beran, P. S.: Assistant Professor, Air Force Institute of Technology, Wright Patterson AFB, OH. Personal Interview. 7 Feb 1994.

18. Beran, P. S. and Morton, S. A.: "Computational Analysis of Hopf-Bifurcation for 2-D Flows With Application to Airfoil Flutter," In Preparation.

19. Bland, S. R. and Seidel, D. A.: "Calculation of Unsteady Aerodynamics for Four AGARD Standard Aeroelastic Configurations." NASA TM 85817, May 1984.

20. Breitbach, E. J. *Flutter Analysis of an Airplane With Multiple Structural Nonlinearities in the Control System.* NASA Technical Paper 1620, March 1980.

21. Buning, P. G. and Steger, J. L.: "Solution of the Two-Dimensional Euler Equations with Generalized Coordinate Transformation Using Flux Vector Splitting." AIAA Paper 82-0971, June 1982.

22. Buxton, B.J. *Comparison of Two Shock Capturing Methods for Calculation of Transonic Airfoil Flutter.* MS thesis, AFIT/GAE/ENY/95D-04. School of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB OH, December 1995.

23. Chakravarthy, S. R. "Euler Equations-Implicit Schemes and Boundary Conditions," *AIAA Journal, Vol. 21, No. 5*, May, 1983.

24. Chandrasekhara, M. and Carr, L.: "Flow Visualization Studies of the Mach Number Effects on the Dynamic Stall of an Oscillating Airfoil," *AIAA 27th Aerospace Sciences Meeting.* AIAA 89-0023, 1989.

25. Chapman, G.T. and Tobak, M.: "Bifurcation in Unsteady Aerodynamics-Implications for Testing." NASA TM-100083, 1988.

26. Chyu, W. J., Davis S. S. and Chang, K. S.: "Calculation of Unsteady Transonic Flow Over an Airfoil," *AIAA Journal, Vol 19, No. 6*, June 1981.

27. Chyu, W. J. and Davis S. S.: "Numerical Studies of Unsteady Flow Over an Oscillating Airfoil," In *AGARD CP 374, Transonic Unsteady Aerodynamics and its Aeroelastic Applications*, January 1985.

28. Chow, L. J. and Goorjian, P. M.: "Implicit Unsteady Transonic Airfoil Calculations at Supersonic Freestreams." AIAA Paper 82-0934, St. Louis, Missouri, June 7-11, 1982.

29. Couston, M.; Angelini, J. J.; and Mulak, P.: "Application de L'equation des petites perturbations transonniques aux calculs d'ecoulements bidimensionnels instationnaires." Rech. Aerosp., No. 1975-5, 1975, pp. 325-340.

30. Clarkson, J. D. *A Computational Investigation of Airfoil Stall Flutter.* MS Thesis. Naval Postgraduate School, Monterey, CA, March 1992 (AD-A247302).

31. Carr, L. W., McAlister, K. W. and McCroskey, W. J.: *Analysis of the Development of Dynamic Stall Based on Oscillating Airfoil Experiments.* NASA TN-D-8382, 1977.

32. Davis, S. D. and Malcolm, G. N.: "Transonic Shock-Wave/Boundary-Layer Interactions on an Oscillating Airfoil," *AIAA Journal, Vol. 18, No. 11*, November, 1980.

33. den Boer, R. G. and Houwink, R.: "Analysis of Transonic Aerodynamic Characteristics for a Supercritical Airfoil Oscillating in Heave, Pitch and With Oscillating Flap." In AGARD CP-374, *Transonic Unsteady Aerodynamics and its Aeroelastic Application*, January 1985.

34. Dowell, E. H., et. al. *A Modern Course in Aeroelasticity.* The Netherlands: Sijthoff and Noordhoff, 1978.

35. Dowell, E.H.: "Eigenmode Analysis in Unsteady Aerodynamics: Reduced Order Models," *36th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, New Orleans, LA, April 10-12, 1995."* AIAA-95-1450.

36. Dowell, E.H.: Dean of Engineering, Duke University, Durham, North Carolina. Personal Interview. April, 1996.

37. Driver, M. J. *High Resolution TVD Schemes for the Analysis of I. Inviscid Supersonic and Transonic Flows, II. Viscous Flows with Shock-Induced Separation and Heat Transfer.* PhD Dissertation. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, November 1991.

38. Edwards, J. W.; Bland, S. R.; and Seidel, D. A.: "Experience with Transonic Unsteady Aerodynamic Calculations." In AGARD CP-374, *Transonic Unsteady Aerodynamics and its Aeroelastic Application*, January 1985.

39. Edwards, J. W.: "Applications of Potential Theory Computations to Transonic Aeroelasticity." Paper No. ICAS-86-2.9.1, *Fifteenth Congress of the International Council for the Aeronautical Sciences*, London, England, September 1986.

40. Edwards, J. W. and Thomas, J. L.: "Computational Methods for Unsteady Transonic Flows," *AIAA 25th Aerospace Sciences Meeting.* AIAA-87-0107, 1987.

41. Edwards, J. W. and Malone, J. B.: *Current Status of Computational Methods for Transonic Unsteady Aerodynamics and Aeroelastic Applications.* NASA TM-104191, 1992.

42. Girodroux-Lavigne, P. and Le Balleur, J. C.: "Calcul D'Ecoulements Instationnaires Transsoniques Avec Decollements Par Interaction Visqueux - Non Visqueux." In AGARD CP-374, *Transonic Unsteady Aerodynamics and its Aeroelastic Application*, January 1985.

43. Granville, P. S. "A Modified Van Driest Formula for the Missing Length of Turbulent Boundary Layers in Pressure Gradients," *Journal of Fluids Engineering, Volume 111*: 94-97, 1989.

44. Greenwood, T. *Principals of Dynamics, 2nd Edition.* Prentice Hall, New Jersey, 1988.

45. Griewank, A. and Reddien, G. "The Calculation of Hopf Points by a Direct Method," *IMA Journal of Numerical Analysis, 3*: 295-303, 1983.

46. Guruswamy, G. P. and Goorjian, P. M.; "Efficient Algorithm for Unsteady Transonic Aerodynamics of Low-Aspect-Ratio Wings." *Journal of Aircraft, Vol. 22, No. 3*, March 1985.

47. Guruswamy, G. P.; Goorjian, P. M.; and Tu, E. L.: "Unsteady Transonics of a Wing with Tip Store." *Journal of Aircraft, Vol. 23, No. 8*, August 1986.

48. Hafez, M. M.; Osher, S.; and Whitlow, W. Jr.: "Improved Finite Difference Schemes for Transonic Potential Calculations." AIAA Paper 84-0092, Los Angeles, California, January 1984.

49. Hafez, M.; Palaniswamy, S. and Mariani P.: "Calculations of Transonic Flows with Shocks Using Newton's Method and Direct Solver, Part II." AIAA Paper 88-0226, *AIAA 26nd Aerospace Sciences Meeting*, Reno Nevada, January 1988.

50. Harten, Ami "High Resolution Schemes for Hyperbolic Conservation Laws," *Journal of Computational Physics, 49*: 357-393 (1983).

51. Hess, R. W.; Seidel, D. A.; Igoe, W. B.; and Lawing, P. L. "Highlights of Unsteady Pressure Tests on a 14 Percent Supercritical Airfoil at High Reynolds Number, Transonic Condition," *Aerospace Sciences Meeting.* AIAA Paper No. 87-0035, Reno, Nevada, January 12-15, 1987.

52. Houwink, R. and van der Vooren, J.: "Improved Version of LTRAN2 for Unsteady Transonic Flow Computations." *AIAA Journal, vol 18*, August 1980.

53. Houwink, R.: "Unsteady Viscous Transonic Flow Computations Using the LTRAN2-NLR Code Coupled with Green's Lag-Entrainment Method." In Numerical and Physical Aspects of Aerodynamic Flows II, Chapter 15, ed. T. Cebeci, Springer, New York, 1984.

54. Houwink, R. and Veldman, A. E. P.: "Steady and Unsteady Flow Computations for Transonic Airfoils." AIAA Paper 84-1618, Snowmass, Colorado, June 1984.

55. Hui, W.H. and Tobak, M.: "Bifurcation Analysis of Aircraft Pitching Motions About Large Mean Angles of Attack." *Journal of Aircraft, Vol. 7, No. 1*, January-February 1984.

56. Isaacson, E. and Keller, H.B. 1966 *Analysis of Numerical Methods.* John Wiley & Sons, New York.

57. Isogai, K.: "Numerical Study of Transonic Flutter of a Two-Dimensional Airfoil." NAL Report TR-716T, July 1980.

58. Jackson, C.P. "A Finite-Element Study of the Onset of Vortex Shedding in Flow Past Variously Shaped Bodies," *Journal of Fluid Mechanics, Volume 182*: 23-45, 1987.

59. Jameson, A. and Venkatakrishnan, V.: "Transonic Flows About Oscillating Airfoils Using the Euler Equations." AIAA Paper 85-1514, *AIAA* 7^{th} *Computational Fluid Dynamics Conference*, 1985.

60. Janus, J. M.: *The Development of a Three Dimensional Split Flux Vector Euler Solver with Dynamic Grid Applications.* M.S. Thesis, Mississippi State, Mississippi, August 1984.

61. Landon, R. H. "NACA 0012. Oscillatory and Transient Pitching," *Compendium of Unsteady Aerodynamic Measurements.* AGARD R-702, 1982.

62. Le Balleur, J. C. and Girodroux-Lavigne, P.: "Prediction of Buffeting and Calculation of Unsteady Boundary Layer Separation Over Airfoils." Presented at IUTAM Symposium on "Boundary Layer Separation," London, University College, August 1986.

63. Le Balleur, J. C.: "Numerical Viscid-Inviscid Interaction in Steady and Unsteady Flows." Proc. 2^{nd} *Symposium on Numerical and Physical Aspects of Aerodynamic Flows*, California State University, Long Beach, California, 1983.

64. Le Balleur, J. C.; and Girodroux-Lavigne, P.: "A Semi-Implicit and Unsteady Numerical Method of Viscous-Inviscid Interaction for Transonic Separated Flows." Rech. Aerosp. 1984-1, 1984, pp. 14-37.

65. Le Balleur, J. C.; and Girodroux-Lavigne, P.: "A Viscous-Inviscid Interaction Method for Computing Unsteady Transonic Separation." Proc 3^{rd} Symposium on Numerical and Physical Aspects of Aerodynamic Flows, ed. T. Cebici, Springer-Verlag, 1986.

66. Lecointe, Y. and Piquet, J.: "Unsteady Viscous Flow Around Moving Circular Cylinders and Airfoils." AIAA Paper 85-1490, *AIAA* 7^{th} *Computational Fluid Dynamics Conference*, 1985.

67. Levy, L. L., Jr.: "Experimental and Computational Steady and Unsteady Transonic Flows about a Thick Airfoil." *AIAA Journal, Vol. 16, No. 6*, June 1978.

68. Levy, L. L., Jr.: "Predicted and Experimental Steady and Unsteady Transonic Flows about a Biconvex Airfoil." NASA TM 81262, Febuary 1981.

69. Keller, H.B. 1977 Numerical Solution of Bifurcation and Nonlinear Eigenvalue Problems. *Applications of Bifurcation Theory.* Academic Press, New York, p. 359.

70. Keller, H.B. 1982 Continuation Methods in Computational Fluid Dynamics. *Numerical and Physical Aspects of Aerodynamic Flows*, (ed. T. Cebeci). Springer-Verlag, New York, p. 3.

71. Keller, H. B. "Bifurcation Theory and Nonlinear Boundary-Value Problems," Lecture Notes California Institute of Technology 1985.

72. Knight, D. and M. Visbal. "The Baldwin-Lomax Turbulence Model for Two-Dimensional Shock-Wave/Boundary-Layer Interactions," *AIAA Journal, 22:* 921-928 (July 1984).

73. Kousen K. A., and Bendiksen, O. O.: "Nonlinear Aspects of the Transonic Aeroelastic Stability Problem," *AIAA* 29^{th} *Structures, Structural Dynamics and Materials Conference, Williamsburg, VA, April 18-20, 1988."* AIAA-88-2306.

74. Kousen K. A., and Bendiksen, O. O.: "Limit Cycle Phenomena in Computational Transonic Aeroelasticity," *Journal of Aircraft, Vol. 31, No. 6*, November-December, 1994.

75. Laganelli, A. L. and Dash, S. M.: *Turbulence Modeling, Interim Report*, SBIR AF91-108. Flight Dynamics Laboratory, Wright Research and Development Center, Wright Patterson AFB OH, July 1990.

76. Lambourne, N. C. *Compendium of Unsteady Aerodynamic Measurements.* AGARD Report No. 702, 1982.

77. Lutton, M. J. *Comparison of C and O-Grid Generation Methods Using a NACA 0012 Airfoil.* MS thesis, AFIT/GAE/ENY/89D-21. School of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB OH, December 1981.

78. Lutton, M. J. *Hopf Bifurcation in Viscous, Low Speed Flows About an Airfoil with Structural Coupling.* PhD Dissertation. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1993.

79. Lutton, M. J. and Beran, P. "Hopf Bifurcation in Viscous, Low Speed Flows About an Airfoil With Structural Coupling," Computers and Fluids Vol 23, No. 2, pp. 323-345, 1994.

80. Mabey, D. G. "Oscillatory Flows from Shock-Induced Separations on Biconvex Aerofoils of Varying Thickness in Ventilated Wind Tunnels," AGARD CP-296, 1980.

81. Mabey, D. G.; Welsh B. L.; and Cripps, B. E. "Periodic Flows on a Rigid 14% Thick Biconvex Wing at Transonic Speeds," RAE Technical Report 81059, May 1981.

82. Magnus, R., and Yoshihara, H.: "The Transonic Oscillating Flap," *AIAA 9th Fluid and Plasma Dynamics Conference, San Diego, CA, July 14-16, 1976."* AIAA-76-327.

83. Magnus, R. J.: *Calculations of Some Unsteady Transonic Flows about the NACA 64A006 and 64A010 Airfoils.* Technical Report AFFDL-TR-77-46, July 1977.

84. Magnus, R. J.: *Some Numerical Solutions of Inviscid, Unsteady, Transonic Flows Over the NLR 7301 Airfoil.* CASD/LVP 78-013, Convair Division of General Dynamics, San Diego, California, January 1978.

85. Malone, J. B. and Sankar, N. L.: "Numerical Simulation of Two Dimensional Unsteady Transonic Flows Using the Full Potential Equation." *Journal of Aircraft, Vol. 22, No. 8*, August 1984.

86. Marstiller, J. W.; Guruswamy, P.; Yang, T. Y.; and Goorjian, P. M.; "Effects of Viscosity and Modes on Transonic Flutter Boundaries of Wings." AIAA Paper 84-0870, Palm Springs, California, 1984.

87. McAlister, K. W. and Carr, L. W.: *Dynamic Stall Experiments on the NACA 0012 Airfoil.* NASA TP-1100, 1978.

88. McCroskey, W. J., McAlister, K. W., Carr, L. W., and Pucci, S. L.: *An Experimental Study of Dynamic Stall on Advanced Airfoil Sections, 3 Volumes.* NASA TM-84245, 1982.

89. McDevitt, J. B.; Levy, L. L.. Jr.; and Deiwert, G. S.: "Transonic Flows about a Thick Circular-Arc Airfoil." *AIAA Journal, Vol. 14, No 5*, May 1976.

90. McDevitt, J. B.: "Supercritical Flow About a Thick Circular-Arc Airfoil." NASA TM-78549, January 1979.

91. McDevitt, J. B. and Okuno, A. F.: "Static and Dynamic Pressure Measurements on a NACA 0012 Airfoil in the Ames High Reynolds Number Facility." NASA Technical Paper 2485, June 1985.

92. Metha, U. B. "Dynamic Stall of an Oscillating Airfoil," *Unsteady Aerodynamics.* AGARD CP-227, 1978.

93. Moran, K. J. PhD Candidate, Air Force Institute of Technology, Wright Patterson AFB, OH. Personal Interview. October 1993.

94. Morton, S. A. and Beran, P. S.: "Nonlinear Analysis of Airfoil Flutter at Transonic Speeds," 13^{th} *AIAA Applied Aerodynamics Conference.* AIAA 95-1905-CP, 1995.

95. Morton, S. A., Beran, P. S.: "Hopf-Bifurcation Analysis of Airfoil Flutter at Transonic Speeds," *AIAA* 34^{th} *Aerospace Sciences Meeting and Exhibit, Reno, NV Jan 15-18, 1996."* AIAA-96-0060.

96. Morton, S. A. *Numerical Simulation of Compressible Vortices.* MS Thesis. Air Force Institute of Technology, Wright-Patterson AFB, OH, 1989.

97. Nayfeh, A.; Mook, D.; Beran, P.S. and Planeaux, J.: *Modern Analysis of Nonlinear Systems With Applications*, AIAA Professional Studies Series Short Course, New Orleans, Louisiana, August 15-16, 1991.

98. Orkwis, P. D.: "A Comparison of Newton's Method Solvers for the Navier-Stokes Equations," AIAA paper No. 92-2644, 1992.

99. Osher, S.; Hafez, M.; and Whitlow, W., Jr.: "Entropy Condition Satisfying Approximations for the Full Potential Equation of Transonic Flow." *Mathematics of Computations, Vol. 44, No. 169*, January 1985.

100. Osswald, G. A.; Ghia, K. N.; and Ghia, U.: "An Implicit Time-Marching Method for Studying Unsteady Flow with Massive Separation." AIAA Paper 85-1489, *AIAA* 7^{th} *Computational Fluid Dynamics Conference*, 1985.

101. Peyret, R. and Taylor, T.D. 1983 *Computational Methods for Fluid Flow.* Springer-Verlag, New York.

102. Press, W. H. et.al. *Numerical Recipes, The Art of Scientific Computing.* Cambridge University Press, 1987.

103. Rivera, J. A.; Dansberry, B.E.; Bennett, R. M.; Durham, M. H.; and Silva, W. A.: "NACA 0012 Benchmark Model Experimental Flutter Results with Unsteady Pressure," AIAA 92-2396, 1992.

104. Rivera, J. A., Jr.; Dansberry, B. E.; Bennett, R. M.; Durham, M. H.; and Silva W. A.. *NACA0012 Benchmark Model Experimental Flutter Results with Unsteady Pressure Distributions* NASA TM-107581, March 1992.

105. Rivera, J. A.; Dansberry, B. E.; Farmer, M. G.; Eckstrom, C. V.; Seidel, D. A.; and Bennett, R. M.: "Experimental Flutter Boundaries with Unsteady Pressure Distributions for the NACA 0012 Benchmark Model," AIAA 91-1010, 1991.

106. Rivera, J. A. et. al. "Experimental Flutter Boundaries with Unsteady Pressure Distributions for the NACA 0012 Benchmark Model," *AIAA* 32^{nd} *Structures, Structural Dynamics, and Materials Conference.* AIAA 91-1010, 1991.

107. Rizzetta, D. P. and Borland, C. J.: "Unsteady Transonic Flow Over Wings Including Inviscid/Viscous Interaction." *Journal of Aircraft, Vol. 21, No. 3*, March 1983.

108. Roe, P. L. "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics, 43*: 357 (1981).

109. Romanowski M.C. and Dowell, E.H.: "Aeroelastic Analysis of an Airfoil Using Eigenmode Based Reduced Order Unsteady Aerodynamics," *36th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, New Orleans, LA, April 10-12, 1995."* AIAA-95-1380.

110. Romanowski M.C. and Dowell, E.H.: "Reduced Order Euler Equations for Unsteady Aerodynamic Flows: Numerical Techniques," *34th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 15-18, 1996."* AIAA-96-0528.

111. Roose, D. and Hlavacek, V. "A Direct Method for the Computation of Hopf Bifurcation Points," *SIAM Journal of Applied Math,* Vol 45, No. 6, pp. 879-894, 1985.

112. Sankar, N. L. and J. C. Wu. "Viscous Flow Around Oscillating Airfoil - A Numerical Study," *AIAA Conference.* AIAA 78-1225, 1978.

113. Sankar, N. L. and Y. Tassa. "Reynolds Number and Compressibility Effects on Dynamic Stall of a NACA 0012 Airfoil," *AIAA 18th Aerospace Sciences Meeting.* AIAA 80-0010, 1980.

114. Sankar, L. M.; Ruo, S. Y.; and Malone, J. B.: "Application of Surface Transpiration in Computational Aerodynamics." AIAA Paper 86-0511, Reno Nevada, January 6-9, 1986.

115. Seydel, R. *From Equilibrium to Chaos: Practical Bifurcation and Stability Analysis.* New York: Elsevier Science Publishing Company, 1988.

116. Shamroth, S. J. "Calculation of Steady and Oscillating Airfoil Flow Fields via the Navier-Stokes Equations," *AIAA 22nd Aerospace Sciences Meeting.* AIAA 84-0525, 1984.

117. Shubin, G. R.; Stephens, A. B.; and Glaz H. M. "Steady Shock Tracking and Newton's Method Applied to One-Dimensional Flow," *Journal of Computational Physics, 39,* 364-374 (1981).

118. Sides, J.: "Computation of Unsteady Transonic Flows with an Implicit Numerical Method for Solving the Euler Equations," Rech. Aerosp. 1985-2.

119. Smith, F.R.: PhD Candidate, Air Force Institute of Technology, Wright Patterson AFB, OH. Personal Interview. April, 1996.

120. Smith, G. E., Whitlow, W., Jr.; and Hassan, H. A.: "Unsteady Transonic Flows Past Airfoils Using the Euler Equations." AIAA Paper 86-1764-CP, San Diego California, June 9-11, 1986.

121. Smith, M.J. *Flight Loads Prediction Methods for Aircraft: Vol I. Euler/Navier-Stokes Aeroelastic Method (ENS3DAE) Technical Development Summary: Version 4.0* (to be published: supersedes WRDC-TR-89-3104, Nov 1989, D.M. Schuster, J. Vadyak, E.H. Alta.),

122. St. Hilaire, A. O.; Carta, F. O.; Fink, M. R.; and Jepson, W. D.: *The Influence of Sweep on the Aerodynamic Loading of an Oscillating NACA 0012 Airfoil, Volume I - Technical Report.* NASA CR-3092, 1979.

123. St. Hilaire, A. O.and Carta F. O.: *Analysis of Unswept and Swept Wing Chordwise Pressure Data From an Oscillating NACA 0012 Airfoil Experiment.* NASA CR-3567, 1983.

124. Steger, J. L.: "Implicit Finite-Difference Simulation of Flow about Arbitrary Two-Dimensional Geometries." *AIAA Journal, Vol. 16, No. 7,* July 1978.

125. Steger, J. L. and Bailey, H. E.: "Calculation of Transonic Aileron Buzz," *AIAA Journal, 18, No. 3,* March 1980.

126. Steinbrenner, J. P., Chawner, J. R., Fouts, C. L. *The GRIDGEN 3D Multiple Block Grid Generation System, Volume 1: Final Report,* WRDC-TR-90-3022. Flight Dynamics Laboratory, Wright Research and Development Center, Wright Patterson AFB OH, July 1990.

127. Steinhoff, J. and Jameson, A.: " Multiple Solutions of the Transonic Potential Flow Equation." *AIAA Journal, Vol. 20, No.11,* November 1982.

128. Stephen, E. , Walker, J., Roh, J., Eldred, T., and Beals, M.. "Extended Pitch Axis Effects on the Flow About Pitching Airfoils," *AIAA 27th Aerospace Sciences Meeting.* AIAA 89-0025, 1989.

129. Theodorsen, T. *General Theory of Aerodynamic Instability and the Mechanism of Flutter.* NACA Report 496, 1935.

130. Theodorsen, T. and Garrick, I. E.: *Mechanism of Flutter: A Theoretical and Experimental Investigation of the Flutter Problem.* NACA Report 685, 1940.

131. Theodorsen, T. and Garrick, I. E.: *Nonstationary Flow About a Wing-Aileron-Tab Combination Including Aerodynamic Balance.* NACA Report 736, 1942.

132. Thibert, J. J., Grandjacques, M., and Ohman, L. H.: "NACA 0012 Airfoil," *Experimental Data Base for Computer Program Assessment.* AGARD AR-138, 1979.

133. Tijdeman, H. *Investigations of the Transonic Flow Around Oscillating Airfoils.*, National Aerospace Laboratory NLR, the Netherlands, NLR TR 770900, 1978.

134. Tjatra, I.W., Kapania, R.K., and Grossman, B.: "Transonic Flutter Analysis of Aerodynamic Surfaces in the Presence of Structural Nonlinearities," *Computing Systems in Engineering, Vol. 1, No.s 2-4, pp. 211-218*, 1990.

135. Tromp, J. C. *The Dependence of the Time-Asymptotic Structure of 3-D Vortex Breakdown on Boundary and Initial Conditions.* PhD Dissertation. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, July 1995.

136. Venkatakrishnan, V.: "Newton Solution of Inviscid and Viscous Problems." AIAA Paper 88-0413, *AIAA 26nd Aerospace Sciences Meeting*, Reno Nevada, January 1988.

137. Vierck, R. K. *Vibration Analysis* (Second Edition). New York: Harper & Row, 1979.

138. Visbal, M. R. *Calculation of Viscous Transonic Flows About a Supercritical Airfoil*, AFWAL-TR-86-3013. Flight Dynamics Laboratory, Air Force Wright Aeronautical Laboratories, Wright Patterson AFB OH, July 1986.

139. Visbal, M.R. "Dynamic Stall of a Constant-Rate Pitching Airfoil," *Journal of Aircraft, Vol. 27, No. 5*:400-407, May, 1990.

140. Viviand, H. "Numerical Solutions of Two-Dimensional Reference Test Cases," *Test Cases for Inviscid Flow Field Methods.* AGARD AR-211, 1985.

141. Whitlow, W., Jr.: "XTRAN2L: A Program for Solving the General-Frequency Unsteady Transonic Small Disturbance Equation." NASA TM-85723, November 1983.

142. Whitfield, D. L. and Taylor, L. K. "Discretized Newton-Relaxation Solution of High Resolution Flux-Difference Split Schemes," *Proceedings of the* 10th *AIAA Computational Fluid Dynamics Conference.* 134-145. Honolulu, HI, June 24-27, 1991.

143. Williams, M. H.; Bland, S. R.; and Edwards, J. W.: "Flow Instabilities in Transonic Small-Disturbance Theory." *AIAA Journal, Vol. 23, No. 10*, October 1985.

144. Wu, J., Kaza, K.R.V., and Sankar, L.N. "A Technique for the Prediction of Airfoil Flutter Characteristics in Separated Flow," AIAA-87-0910, 1987.

145. Wu, J. C., Srivastava, R., and Sankar, L. N.: "Application of Navier-Stokes Analysis to Stall Flutter," *Lewis Structures Technology - 1988, Volume 1.* NASA CP-3003, 1988.

146. Wu, J.C., Kaza, K.R.V. and Sankar,L.N.: "A Technique for the Prediction of Airfoil Flutter Characteristics in Separated Flow," *28th AIAA Structures, Structural Dynamics and Materials Conference, New York, NY, 1987."* AIAA-87-0910.

147. Yang, Z. C. and Zhao, L. C.: "Analysis of Limit Cycle Flutter of an Airfoil in Incompressible Flow," *Journal of Sound and Vibration, 123:* 1-13, 1988.

148. Yee, H. C. *Implicit Total Variation Diminishing (TVD) Schemes for Steady-State Calculations.* NASA TM-84342.

149. Yee, H. C. *A Class of High Resolution Explicit and Implicit Shock Capturing Methods.* NASA TM-101088, (February 1989).

150. Yee, H. C. *Application of Second-Order-Accurate Total Variation Diminishing (TVD) Schemes to the Euler Equations in General Geometries.* NASA TM-85845.

151. Young, D. M. and Gregory, R. T. *A Survey of Numerical Methods, Volume I.* New York: Dover Publications Inc., 1988.

152. Zimmerman, H. *The Application of Transonic Unsteady Methods for Calculation of Flutter Airloads.* In AGARD CP-374 "Transonic Unsteady Aerodynamics and its Aeroelastic Applications," January 1985.

153. Zucrow, M. J. and Hoffman, J. D. *Gas Dynamics, Volume I.* New York: John Wiley & Sons, 1976.